

NOŠITEL
VYZNAMENÁNÍ
ZA BRANNOU
VÝCHOVU
I. A II. STUPNĚ



ŘADA B PRO KONSTRUKTÉRY

ČASOPIS PRO ELEKTRONIKU
A AMATÉRSKÉ VYSÍLÁNÍ
ROČNÍK XXXII/1983 ●● ČÍSLO 1

V TOMTO SEŠITĚ

| | |
|--|----|
| Polytechnická výchova ve Svazarmu | 1 |
| MIKROPROCESORY a MIKROPOČÍTAČE | |
| I. Úvod | 2 |
| Mikroprocesory? Proč? | 2 |
| Mikroprocesor? Jak vlastně pracuje? | 3 |
| Co potřebujeme k práci s mikroprocesory? | 5 |
| Tester TST-01 a přípravy TST-02, 03 | 7 |
| Simulátor EPROM | 8 |
| II. Amatérský osobní mikropočítač Intelka | 10 |
| Popis zapojení | 10 |
| Osazení desek | 15 |
| Oživení desek | 15 |
| III. Mikropočítačový systém | |
| JPR-1 | 22 |
| Sběrnice ARB-1 | 23 |
| Popis signálů sběrnice ARB-1 | 27 |
| Konstrukce a oživení sběrnice ARB-1 | 31 |
| Deska procesoru JPR-1 | 31 |
| Konstrukce systému JPR-1 | 31 |
| Blokové schéma desky procesoru | 31 |
| Schéma zapojení desky JPR-1 | 32 |
| Paměti, porty | 35 |
| Oživení desky procesoru | 37 |
| Oprava (pro AR B5/82) | 40 |

AMATÉRSKÉ RADIO ŘADA B

Vydává ÚV Svazarmu ve vydavatelství NAŠE VOJSKO, Vladislavova 26, 133 66 Praha 1, tel. 26 06 51-7. Šéfredaktor ing. Jan Klabal, redaktor Luboš Kalousek, OK1FAC. Redakční rada: RNDr. V. Brunnhofer, V. Brzák, K. Donát, V. Gazda, A. Glanc, I. Harminc, M. Háša, Z. Hradský, P. Horák, J. Hudec, ing. J. T. Hyan, ing. J. Jaroš, doc. ing. dr. M. Joachim, ing. F. Králík, RNDr. L. Kryška, J. Kroupa, ing. E. Möckl, V. Němec, RNDr. L. Ondříš, CSc., ing. F. Smolík, ing. E. Smutný, ing. V. Teska, doc. ing. J. Vackář, laureát st. ceny KG, J. Vorlíček.

Redakce Jungmannova 24, 113 66 Praha 1, tel. 26 06 51-7, šéfredaktor linka 354, redaktor linka 353, sekretářka linka 355. Ročně vyjde 6 čísel. Cena výtisku 5 Kčs, pololetní předplatné 15 Kčs. Rozšiřuje PNS, v jednotkách ozbrojených sil vydavatelství NAŠE VOJSKO, administrace Vladislavova 26, Praha 1. Objednávky přijímá každá pošta i doručovatel. Objednávky do zahraničí využívá PNS, ústřední expedice a dovoz tisku, závod 01, Kafkova 9, 160 00 Praha 6. Tiskne NAŠE VOJSKO, n. p., závod 08, 162 00 Praha 6, Vlastina 710.

Za původnost a správnost příspěvku odpovídá autor. Návrhy v redakci a telefonické dotazy pouze po 14. hodině. Číslo indexu 46 044.

Toto číslo má vyjít podle plánu 19. 1. 1983.

© Vydavatelství NAŠE VOJSKO

Polytechnická výchova, zejména mládeže, se v době vědeckotechnické revoluce stává nezbytnou, nemá-li se trvale prohlubovat rozpor mezi ekonomickými potřebami společnosti a jejich zajišťová-

„Polytechnická výchova ve Svazarmu“

Na tomto zasedání byla řešena řada závažných a podnětných myšlenek, zaměřená na zlepšení a zkvalitnění pracovní výuky mladé generace, která by v ještě větší míře přispívala k poznávání a osvojování si základů techniky, modernímu technickému myšlení a tím napomáhala k výraznějšímu vědeckotechnickému rozvoji celé společnosti. Polytechnická výchova ve Svazarmu má již dlouholetou tradici, která vychází především z:

- výchovy k aktivnímu vztahu k práci pro společnost,
- přípravy občanů a zejména mládeže k úkolům vědeckotechnického rozvoje,
- vytváření aktivního vztahu k technice a rozvíjení tvořivého myšlení,
- propagování technických povolání u mládeže a dětí,
- prohlubování technických znalostí a dovedností.

Získané zkušenosti ukázaly, že tento přístup k plnění koncepce rozvoje jednotlivých odborností vytváří širší prostor k uspokojování zájmů a zálib jednotlivců i kolektivů v souladu s potřebami výstavby a obrany socialistické společnosti, tak jak to vyžaduje politika KSČ. Zprávu, která si vzala na tomto plenárním zasedání za cíl všestranně posoudit, jak branná organizace ve smyslu závěrů XVI. sjezdu KSČ a rezoluce VI. sjezdu Svazarmu v oblasti polytechnické výchovy mládeže plní stanovené úkoly, přednesl s. generálporučík ing. J. Činčár, z jehož referátu vyjímáme:

„Při hodnocení dosavadního stavu můžeme konstatovat, že koncepce podnikly do většiny základních organizací a že se stávají programem práce jejich klubů a kroužků. Avšak stále se setkáváme s tím, že koncepce nejsou naplňovány komplexně, že se naše odbornosti a kluby soustřeďují na realizaci jen vybraných částí, které jsou pro ně schůdnější.“

Svazarm je od svého založení velkou školou polytechnické výchovy, rozvíjející se na široké bázi dobrovolné zájmové činnosti, která je již ve své podstatě spjata s technikou a jejím využíváním. To platí zvláště v odbornostech motorismu, elektronice a letectví.

Naše zkušenosti s dospělými členy i mládeží potvrzují, že výsledky práce závisejí na tom, jak se nám daří koordinovanou činnost působit ve všech sférách, tj. ideově politické, branně technické a branně sportovní.

Celým procesem polytechnické výchovy ve Svazarmu prostupuje politicko-výchovná práce. Aktuálním úkolem zůstává překonávání nekritického hodnocení techniky kapitalistických států a podceňování naší a sovětské techniky u části mládeže, ale i některých našich členů a funkcionářů.

Tato problematika byla obsahem technické propagandy. V jejím objasňování sehrál významnou roli také svazarmovský tisk. Vyzvedneme v tomto směru především práci Světa motorů, Amatérského radia, Modeláře i dalších svazarmovských časopisů. Na dva milióny našich čtenářů si prostřednictvím svazarmovského tisku prohlubují jednak technické znalosti, jednak jsou vedeni i ke konkrétním technickým činnostem. V našich časopisech jsou již pravidelně zařazovány rubriky k rozvoji polytechnické výchovy. Dosavadní výsledky v této oblasti nás však doposud neuspokojují. Svazarmovským časopi-

ním. Nutnost dalšího zkvalitňování odborné výuky a výchovy zaměřené na zvládnutí současné techniky se proto stala hlavní náplní 10. pléna ÚV Svazarmu, které zasedalo 15. října 1982 pod heslem:

sům se zatím nedaří v potřebné míře zobečňovat zkušenosti z práce našich základních organizací a tak napomáhat dalšímu masovému rozvoji a propagaci zájmových branně technických činností.

Rozvoj elektroniky se stále větší měrou podílí na vědeckotechnickém rozvoji. Má mimořádný význam pro obranu naší vlasti a celého socialistického společenství. Na důležitost rozvoje elektroniky poukázal i XVI. sjezd KSČ. Tomuto významu však ještě neodpovídá úroveň rozvoje elektroniky ve Svazarmu.

Radioamatérství patří od počátku k progresivním oborům, jeho členové se vyznačují vysokým stupněm technických znalostí a dovedností. Nízká organizovanost - 3,5 % - je ale limitujícím faktorem efektivnějšího rozvoje polytechnické výchovy. Elektroakustika a videotechnika se svými přehlídkami Hifi-Ama se staly nejen veřejnými přehlídkami technické tvořivosti, ale i místem výměny zkušeností a hledání nových podnětů pro práci.

K obsahové náplni svazarmovských časopisů v blízké budoucnosti vyzdvihl ve své zprávě s. genpor. Činčár zejména potřebu zvýšení pozornosti k mládeži a to již od věku 11 či 12 let, kde je třeba, aby část technicky a konstrukčně zaměřených článků byla srozumitelná právě jí.

Na čtenářích je, aby si plně uvědomili, že AR jako časopis ÚV Svazarmu bude i nadále jako doposud uveřejňovat tzv. „jednoduché konstrukce pro mládež“, aby tak plnilo poslání, které má ve výchově mladé generace. A zde právě občas naráží u technicky vyspělých čtenářů na určitou nevolu.

Někteří nám vytýkají, že proto, že máme výsadní postavení jako jediný konstrukční časopis zaměřený na elektroniku, si nemůžeme dovolit tisknout takové návody jako je „neposedné“ světýlko (AR 4/82), či měřidlo chudého radioamatéra (AR 7-8/82), či za zbytečné ničení papíru pokládají nedávnou „kuchařku pro méně pokročilé“ (AR B5). Jsou to sice dopisy ojedinelé, psané výlučně velmi náročnými čtenáři a jediné z řad vysokoškoláků (čehož si v redakci vysoce ceníme), ale právě na nich je, aby si uvědomili, že nejméně tři čtvrtiny čtenářů je z řad amatérů, a ke zmíněnému AR B5 je poměr došlých dopisů zhruba jedna ku pěti v jeho prospěch. Na ukázkou citujeme z několika: takovou „kuchařku“ častěji, grafická úprava plně vyhovuje, velmi milé překvapení, vysoce kladně hodnotím, počinek, zdařilý, potřebný, pokračujte, jině takové informace neseženu, přijatelná forma pro nejširší veřejnost, něco takového již dávno postrádáme, velký přínos pro nás mírně pokročilé... a tak bychom mohli citovat z mnoha dalších, kteří tento experiment redakce chválí. Je třeba chápat, že na 40 stranách časopisu, který je v řadě A vydáván v nákladu 118 tisíc výtisků a v řadě B 88 tisíc výtisků, budou-li 2 listy pro každého zkušeného čtenáře přínosem, je to dostatečně velký úspěch.

Ve směrech dalšího rozvoje polytechnické výchovy ve Svazarmu (10. plénum ÚV) se upozorňuje na to, že je třeba usměrnit obsah svazarmovských časopi-

MIKROPROCESORY A MIKROPOČÍTAČE

I. Úvod

Ing. Eduard Smutný

Když jsme před rokem rozhodovali, co bude náplní tohoto čísla AR řady B, bylo nám to všem naprosto jasné. Postavíme amatérský osobní počítač a potom popíšeme jeho konstrukci, oživení, programování a bude AR pro konstruktéry, kteří se problematikou mikropočítačů zabývají nebo mají o tento nový obor amatérské činnosti zájem. Kolektiv vedoucích i členů kroužků kybernetiky z Městské stanice MT v Praze se okamžitě pustil do práce.

Nejprve se hledala vhodná mechanická konstrukce a našla se. Ve výprodeji se objevily kalkulačky ELKA, které byly postaveny ještě na tranzistorech a měly proto poměrně velkou desku s plošnými spoji a dobrý konektor. Našel se i název, sloučením původního názvu kalkulačky a názvu firmy, která sehrála pionýrskou roli v oblasti mikropočítačů ve světě, vznikl název INTELKA. Potom jsme začali přemýšlet o sběrnici. Sběrnice jsou vlastně vodiče, které propojují desky počítače nebo mikropočítače; tak, aby bylo možno mikropočítač jednoduše rozšiřovat o další paměti, připojovat nová přídatná zařízení atd. Sběrnice má oproti klasické „kabeláži“ tu výhodu, že není nutné při připojování nové desky zapojovat další svazky vodičů. Při diskusích o tom, jaké signály budou na sběrnici, jakou budou mít funkci a jaké obvody budou tyto signály vysílat a přijímat, však nastaly první problémy. Z části byly zaviněny chudou součástkovou základnou, která je u nás k dispozici pro mikropočítače. Nemyslím tím vlastní mikropočítače, ty by bylo možné si přivést ze zahraničí, ale v této fázi nám dělaly problémy obvody, přímo spolupracující se sběrnici. Jeden mikropočítač a několik dalších obvodů je možné si opatřit, ale není možné počítat se zahraničními součástkami v obvodech, které se opakují na každé desce. Další problémy pramenily z toho, že konektor z ELKY měl pouze 41 špiček a to bylo k realizaci sběrnice z osvědčených amatérských i profesionálních systémů málo. Největší problémy však vznikly z malé zkušenosti členů našeho kolektivu, z nedostatečného názvosloví, z nedostatku literatury a také z nejasností cílů, kterých jsme chtěli dosáhnout. Kdyby naším jediným cílem bylo postavit si amatérský osobní počítač, bylo by nejschůdnější vzít vzor, třeba populární TRS-80 a postavit počítač podle něj. Naším cílem však bylo porozumět celé problematice a především použít perspektivní československou součástkovou základnu, představovanou obvodem TESLA 8080A (tzn. nepoužít obvod Z80, který je ve většině zahraničních osobních počítačů). I když zájem amatérů o obvody 8080A bude jistě pro výrobce z obchodního hlediska málo zajímavý, chtěli jsme naučit členy kroužků jistě kázní v dodržování dohodnuté sou-

částkové základny (až jednou nastoupí do zaměstnání, budou muset dělat z toho, co je, a nezatěžovat stát nároky na devizové prostředky). Další problémy vznikaly z nedostupnosti přídatných zařízení, nikdo si také hned neuvědomoval, jak složitý je řadič floppy disku nebo obvody pro „refresh“ dynamických pamětí, jak budeme programovat a mazat paměti EPROM a další a další věci. Dnes po roce musíme přiznat, že takový osobní počítač INTELKA, jak jsme si ho představovali, dnes ještě nemáme. Máme však zkušenosti, víme o problémech a hledáme cesty, jak je řešit. Víme také, že amatérská stavba osobního počítače je sice reálná, ale bez pomoci profesionálů, jejich měřicích přístrojů, jejich možnosti výroby desek s plošnými spoji apod. je prakticky nemožná. V budoucnu bude třeba určitě podpory celé naší elektroniky zájemcům z řad Svazarmu, škol nebo pionýrských domů, aby mohli v oboru mikropočítačů zájmovou činnost rozvíjet. Že se tato podpora naší elektronice plně vyplácí, je snad každému jasné. Ti pracovníci, kteří u nás zaváděli televizní techniku, mi jistě potvrdí, jak obrovským přínosem je široká základna amatérů, jejichž koníčkem je pak i jejich vlastní zaměstnání. Naše zkušenosti po několikaleté práci s mládeží v oboru číslicové a mikropočítačové techniky ukazují, že chlapci od věku deseti let jsou schopni pochopit činnost hradla, klopného obvodu a základy logiky. Od věku čtrnácti let jsou schopni konstruovat jednoduchá zařízení z integrovaných obvodů podle vlastních nápadů. Mají-li v tomto věku dostatek literatury a někoho, kdo jim poradí, jsou schopni navrhovat mikropočítačové systémy, programovat, navrhovat plošné spoje. V období, kdy se připravují k maturitě, mají pak znalosti dostatečné pro návrh i stavbu počítače. V oblasti programování je situace obdobná. Chlapci a děvčata jsou schopni programovat v jazyku BASIC již od šesté třídy. Kde však mají vzít počítače, literaturu a kdo jim v jejich práci poradí? Jediným možným řešením tohoto problému je zavést osobní počítače do školství a do zájmové činnosti a vytvořit podmínky pro stavbu počítačů v organizacích Svazarmu a v kroužcích při školách, domech pionýrů nebo při větších podnicích našeho elektronického průmyslu.

Základním problémem rozvoje mikropočítačů u nás je nedostatek kvalitní literatury. Dokonce i tak základní věc, jakou je dobrý katalog integrovaných obvodů, na tuzemském trhu není. Nebo si snad někdo myslí, že podle několika u nás publikovaných údajů, třeba u obvodu UART MHB1012, jde něco navrhnout? Jistě že ne. Je nutné půjčit si od někoho a okopírovat pár listů z katalogu Texas Instruments, kde je ekvivalentní obvod TMS6011, teprve potom lze něco dělat. A co teprve, když potřebujeme definici sběrnice MULTIBUS nebo popis formátu záznamu na pružný disk! Odhaduji, že

zajištění nezbytné literatury zabere vývojovému pracovníkovi asi 20 % a amatérovi až 50 % času, který věnuje práci na zařízení. Pokusili jsme se proto uspořádat toto číslo AR tak, aby v něm bylo co nejvíce informací potřebných pro práci s mikropočítačem MHB8080A.

A nyní k uspořádání obsahu tohoto čísla AR. V Městské stanici mladých techniků v Praze vznikají v současné době dva projekty: amatérský osobní počítač INTELKA a profesionálně řešená jednotka programového řízení JPR-1. INTELKA je navrhována a konstrukčně řešena členy kroužku mikropočítačů a lze ji postavit amatérsky, tzn., že její deska s plošnými spoji nepotřebuje prokovovat díry a není tak náročná na zhotovení.

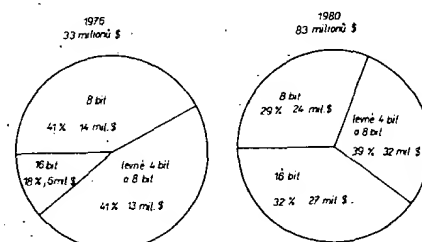
JPR-1 byla řešena v profesionálních podmínkách TESLA Elstroj. Mikropočítač JPR-1 bude používán např. pro řízení technologických zařízení pro výrobu polovodičových součástek. Přidáním mnoha stovek hodin práce doma a za pomoci kolektivu z práce i ze ZO Svazarmu 4006/602 se podařilo vytvořit mikropočítačový systém s procesorem JPR-1. Tento systém vyžaduje ovšem desky s plošnými spoji s prokovenými dírami, vyrobenými v tzv. IV. třídě přesnosti.

Oba systémy mají společné to, že využívají perspektivní součástkové základny, představované v tomto případě mikropočítačem MHB8080A. Díky tomu mají i podobné programové vybavení. Na příkladu těchto dvou systémů chceme ukázat, jak se s mikropočítači dělá, co je k tomu třeba a co musíme znát, abychom mohli navrhnout aplikaci mikropočítačů a psát pro ně programy.

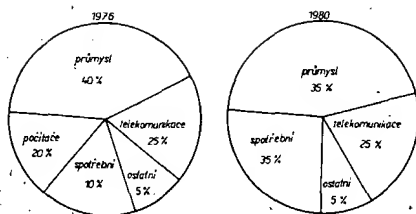
Toto číslo AR je rozděleno na tři části. První slouží jako úvod do celé problematiky, v druhé se dozvíte o konstrukci amatérského osobního počítače INTELKA, ve třetí se seznámíte s mikropočítačovým systémem JPR-1. Doufáme, že toto číslo AR/B pomůže těm, kteří chtějí s mikropočítači něco dělat a nevědí, jak na to.

Mikroprocesory? – Proč?

Mikroprocesory oslavily letos své desáté narozeniny. Těžko bychom hledali v oblasti techniky prvek, který by doznal tak rychlé obliby a miliónové výroby jako mikropočítač a jeho další podpůrné mikroelektronické obvody (obr. 1 až 5). Většina nových prvků prožívá počáteční období nedůvěry, a pak teprve slaví úspěchy, nebo musí překonávat neúspěchy. Mikroprocesory získaly oblibu ihned a pronikly do všech možných oblastí lidského snažení. I když se zpočátku zdály



Obr. 1. Světový trh mikroprocesorů v roce 1976 a 1980 [1]

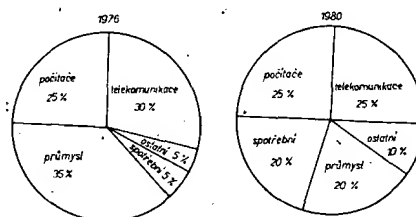


Obr. 2. Rozdělení trhu levných mikroprocesorů [1]

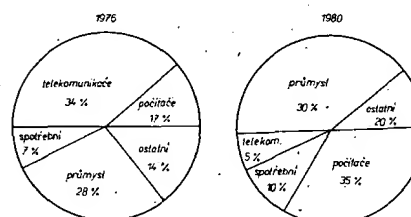
| Rok | Hračky a hry | Kalkulačky | Sporáky | Automobily | Osobní počítače |
|--------|--------------|------------|---------|------------|-----------------|
| 1976 | 6,0 | 1,6 | 0,7 | — | 0,05 |
| 1978 | 8,0 | 5,0 | 3,0 | 1,0 | 0,2 |
| 1980 | 17,0 | 10,0 | 6,0 | 3,0 | 0,6 |
| 1985 * | 50,0 | 25,0 | 10,0 | 10,0 | 1,5 |

* výhled počty v mil.kusů

Obr. 3. Mikroprocesory ve spotřebním zboží na světě [1]



Obr. 4. Rozdělení trhu 8bitových mikroprocesorů [1]



Obr. 5. Rozdělení trhu 16bitových mikroprocesorů [1]

pozice velkých počítačů v oblasti zpracování dat neotřesitelně, ukazuje se, že mikroprocesorové systémy jim dnes tvoří rovnocennou konkurenci. Nakonec i firma IBM musela přijít na trh s osobním mikropočítačem. Položme si otázku, proč slavi mikroprocesory takové úspěchy? Jednou z hlavních příčin je, že aplikace mikroprocesorů nevyžaduje, alespoň v začátku, žádné velké změny technologie. Mám tím na mysli, že se s nimi pracuje stejně jako s jinými integrovanými obvody. Řešitelné bylo na mikroprocesorech umístění centrální jednotky počítače do jednoho pouzdra, avšak vlastní technologie montáže obvodů do desek s plošnými spoji se nezměnila. Další předností mikroprocesorů je, že se s nimi dá pracovat jak při minimálním vybavení vývojového pracoviště, tak i s nejlepšími osciloskopy, analyzátory nebo vývojovými systémy. To umožnilo rozšířit mikroprocesory i do zařízení vyráběných malými firmami a nakonec i mezi amatéry.

Většina nových prvků vyžaduje radikální změnu myšlení mnoha pracovníků pracujících v určitém oboru. Přechod od elektronek k tranzistorům byl jedním z příkladů a trvalo dost dlouho, než se stal tranzistor běžnou záležitostí. I když se dnes říká, že mikroprocesory vyžadují obdobné radikální změny, není to tak zcela pravda. V zařízení nebo přístroji je mikroprocesor jen jeden a ostatní obvody jsou běžné obvody TTL nebo CMOS. I paměť (nebo port) není vlastně nic jiného

než registr. Mikroprocesory se prostě „trefily“ do směru myšlení lidí, zejména mladých. Nechce-li někdo dělat z celé věci vědu, není pro něj složité se naučit mikroprocesory používat, aplikovat a psát pro ně programy. Překonáme-li první zvědavost, kdy chceme vědět, co se vlastně děje uvnitř mikroprocesoru a soustředíme-li se na řešení aplikací a systémových problémů, máme první krok za sebou.

Většinu těch, kteří chtějí s mikroprocesory pracovat, odrazuje nesmírné množství typů mikroprocesorů, pamětí, podpůrných obvodů a řadičů, popř. mají mylnou představu o nutném vybavení přístroji a vývojovými systémy. Z vlastních zkušeností mohu potvrdit, že většina zahraničních obrazkových terminálů je postavena na obvodech TTL a nepoužívá obvody typu 8275. Dále bych chtěl připomenout, že největší práce pro osciloskop je u mikropočítačů tam, kde pracujeme s obvody TTL – pro oživení mikropočítače se osciloskop prakticky nehodí. Další zkušenost je taková, že mikropočítače prostě „chodí“, že s nimi nejsou velké problémy. Proto se také tak rozšířily. Na druhé straně je nutné si uvědomit, že se složitost vývoje, měření, testování a navrhování obvodů přenesla do technologie a k výrobci mikroelektronických součástek a další potřebné součástkové základny. Vzpomínáte si, jakou práci dalo vyvinout a postavit tranzistorový operační zesilovač nebo napájecí zdroj? A uvažuje dnes někdo při aplikaci MAA741 o tom, jak složité je to obvod? Stejně je tomu tak nebo tomu tak bude s mikroprocesory.

Proč vlastně vznikl mikroprocesor? Prvním předpokladem jeho vzniku byl pokrok v technologii, zejména v litografii, která umožnila umístit na čip více než 1000 tranzistorů. Návrh logických systémů pro řízení strojů, měření, sběr dat a informací vyžadoval propojovat logické obvody vždy podle nových požadavků na funkci zařízení. To vyžadovalo zpracovat podklady pro nové desky s plošnými spoji, kabeláž, testování a servis. V určitém období se pro plnění některých funkcí začaly používat minipočítače, které však byly pro menší aplikace drahé a rozměrné. Potom se začaly používat mikroprogramovatelné automaty, pracující sekvencně podle programů, uložených v pamětech PROM. Další cesta vedla k výrobě zákaznických obvodů, speciálně navržených pro plnění potřebné funkce. Ukázalo se však, že (kromě obvodů pro kalkulačky) je složité zadat takovou výrobní sérii jednoho obvodu, aby byl výsledek ekonomicky výhodný. Proto, jakmile to technologie jen trochu dovolila, zrodily se centrální jednotky minipočítačů na jednom čipu – mikroprocesory. Mikropočítač neumí nic více a nic méně než minipočítač a jeho přednost spočívá v tom, že je malý, levný, spolehlivý a energeticky nenáročný. Základním principem počítače je zpracování dat – informací podle programu uloženého v paměti. Chceme-li od něj jinou funkci, stačí zadat jiný program a připojit jiná vstupní a výstupní zařízení. Samozřejmě, že bylo nutné a žádoucí vyvinout i další obvody. Těžko by mohl mít levný mikroprocesor uložený program v drahé feritové paměti. Klíčem k hromadnému rozšíření programovatelných, univerzálních obvodů – mikroprocesorů byl zrod nového typu paměti – paměti typu EPROM. Jak uvidíte sami, nelze napsat program, aniž by v něm nebyla chyba, nebo aniž by to nebylo nutno později měnit. Paměti EPROM jsou paměti, do nichž lze program uložit a libovolně ho používat. Potom je možno obsah paměti vymazat a „nahrát“ nový program. První mikroprocesorové systémy, pracující tře-

ba s mikroprocesorem 8008, měly vlastně pouze tři nové prvky: mikroprocesor, paměť RAM a paměť EPROM. Jinak byly postaveny z běžných TTL nebo MSI obvodů MOS (UART), které se používaly již u minipočítačů. Právě díky tomuto plynulému přechodu od „mini“ k „mikro“ počítačům nenastaly u uživatelů velké potíže při aplikacích mikropočítačů.

U nás je situace o trochu horší, neboť naše minipočítače byly a jsou příliš „velké“ a drahé a nemohly se proto rozšířit tak, jako třeba minipočítač PDP-8, který byl v Evropě nasazen v počtu několika stovek tisíc v měřicích a řídicích systémech. Stejně tak se přeskočilo důležité období automatů, pracujících mikroprogramově podle programu v pamětech PROM. Je zajímavé, že tato technika přežila i mikroprocesory. Vznikly obvody FPLA a PLA a řídicí systémy některých strojů a přístrojů vyjdou s těmito obvody levněji, než s mikroprocesory. Obvody PLA se nyní připravují do výroby i v k. p. TESLA Rožnov. Ukázalo se tedy, že mikroprocesory nejsou radikálně novou technikou pro návrhář elektronických systémů, ale novým stupněm v kontinuitě světového vývoje součástek. Prokazatelně se přednosti mikroprocesorů ukázaly tam, kde se sleduje ekonomie výroby při zvyšování technické úrovně výrobků.

Jednou z určitých nevýhod mikroprocesoru je to, že chceme-li ho použít v malém levném zařízení, musí být buď levný nebo musíme vymyslet pro zařízení další nové funkce, které vyrovnají (pro uživatele) zvýšení ceny zařízení. Někdy je tento problém těžko řešitelný a proto se v zahraničí značně rozšířily čtyř a osmibitové mikropočítače, které mají na čipu i paměť RAM a PROM. Některé typy těchto mikropočítačů se vyrábějí v několika verzích, například s výstupy pro přímé buzení displejů, snímání signálů z klávesnic apod. V naší perspektivní součástkové základně jsou zařazeny obvody 8048 a 8035. Aplikace těchto jednočipových mikropočítačů však vyžaduje znalosti a zkušenosti získané prací s mikroprocesory typu 8080A, neboť u nich se dá ještě leccos proměřit, změnit nebo rozšířit. Práce s jednočipovými mikropočítači nás čeká, a proto je nutné rychle dohnat zpoždění, které vzniklo nedocenením minipočítačů, především toho faktu, že zkušenosti získané s aplikací minipočítače (třeba JPR-12) jsou nejlepší přípravou pro práci s mikropočítači.

[1] Mc-Glynn, Daniel, R.: Modern microprocessor system design. John Wiley & Sons: New York 1980.

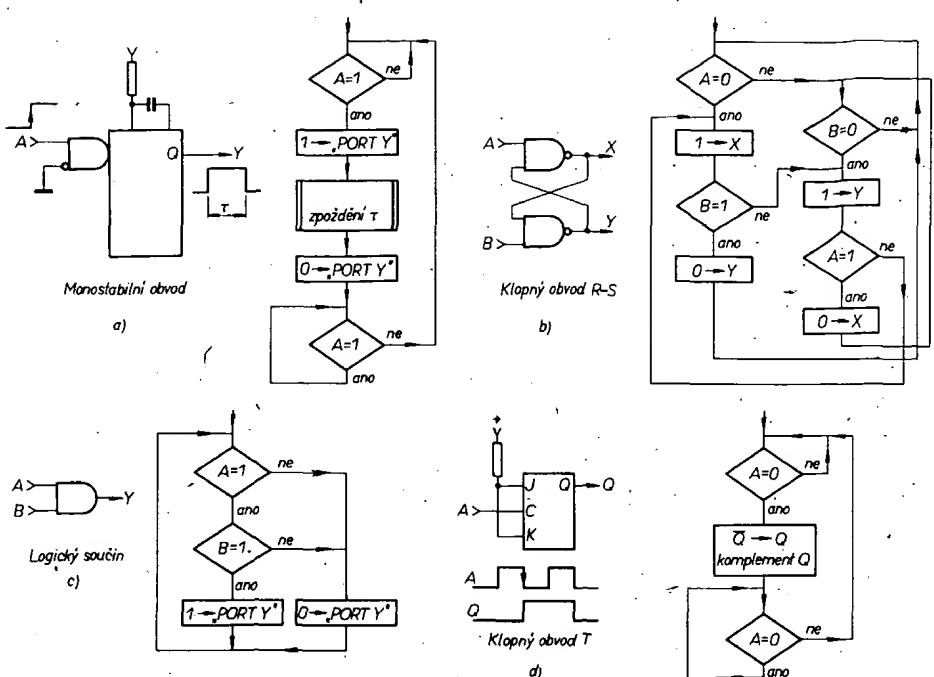
Mikroprocesor? Jak vlastně pracuje?

Mikroprocesor je vlastně centrální jednotka počítače, která se díky složité technologii vejde do jednoho pouzdra integrovaného obvodu. Doplníme-li mikroprocesor o paměti a obvody vstupu a výstupu, dostaneme mikropočítač. Hned z počátku je nutné si uvědomit, že minipočítač a mikropočítač jsou běžné počítače. Předpony „mini“ nebo „mikro“ vyjadřují pouze rozměrové, cenové a technologické úrovně. Dokonce i velké počítače, jak je známe z výpočetních středisek, pracují na zcela stejných principech. Chceme-li pracovat s mikroprocesory, musíme nejprve princip počítače pochopit. Přiznám se, že pochopit, jak pracuje počítač, mi trvalo několik let. Jednak to skutečně není

zcela jednoduché, a jednak bylo tehdy velmi málo literatury. Vystudoval jsem radiotechniku a jsem tedy v tomto oboru vlastně samouk. Až o hodně později jsem zjistil, že klíčem k pochopení činnosti počítače je pochopit, jak pracuje tzv. „čítač programu“, který je v každém počítači.

Každý počítač pracuje na principu Von Neumanna, tzn., že vykonává program, který je uložen v jeho paměti. Program je v paměti zapsán jako sled jednotlivých příkazů, kterým říkáme instrukce. Vykonání jedné instrukce trvá dnešním počítačem asi 1 až 10 μ s, tzn., že realizují za sekundu asi 10⁵ až 10⁶ operací. Z důvodu, které zatím nejsou důležité, vykonává počítač skutečně jednu instrukci za druhou a nikdy se nezastaví. I když mají počítače instrukce i pro zastavení, používají se pouze při opravách stroje nebo při ladění programů. Typický program mikropočítače má asi 4000 instrukcí, kdyby je mikropočítač vykonával jednu za druhou, trvalo by mu to asi 4 ms, tj. přibližně dobu, po kterou se tužka dotýká stolu, když s ní klepneme.

Jak je možné, že program je v mikropočítači několik let a pořád se vykonávají instrukce a pořád je jich dostatek? Jaký obvod vlastně řídí sled vykonávání instrukcí? Je to právě čítač programu. U mikropočítačů typu 8080A, které mají 16bitovou adresu, si ho můžeme představit jako čtyři obvody MH74193, spojené za sebou (obr. 6). Tento čítač je ovládán třemi řídicími signály: NULO VÁNÍ (CLEAR), +1 (COUNT UP) a PARALELNÍ NAHRÁVÁNÍ (LOAD). A teď pozor, základním jevem u počítačů je, že čítač programu obsahuje adresu, z níž bude přečtena následující instrukce. Zapneme-li počítač, čítač programu se vynuluje signálem NULO VÁNÍ a první instrukce bude vzata z adresy 0000 0000 0000 0000. Řídicí obvody počítače, kterým jako celek říkáme řadič, přečtou první instrukci a aritmeticko-logické obvody, kterým říkáme aritmeticko-logická jednotka, tuto instrukci provedou. Při provádění instrukce se signálem +1 zvýší obsah čítače programu o jedničku. Další instrukce se tedy přečte z adresy 0000 0000 0000 0001. Je samozřejmě možné napsat program počítače tak, aby se vykonávala jedna instrukce za druhou – a když by se vykonalo 64 000 instrukcí, čítač programu by se zvýšením o jedničku vlastně vynuloval (přetekl by) a program by probíhal znovu od začátku. Takový program by měl pouze demonstrační význam. Mohl by např. rozsvěcovat nějaké žárovky na vánočním stromku, anebo ovládat motorek někde na výstavě, kde chceme, aby se motorek točil celý den stejně. Na takové úlohy však počítače nepotřebujeme. Ať již něco počítačem řídíme nebo i když zpracováváme data, vždy potřebujeme, aby se provádění programu měnilo podle výsledků vnitřních operací počítače nebo podle stavu signálů, přicházejících na vstupy počítačového



Obr. 7. Náhrada funkce logických obvodů programem

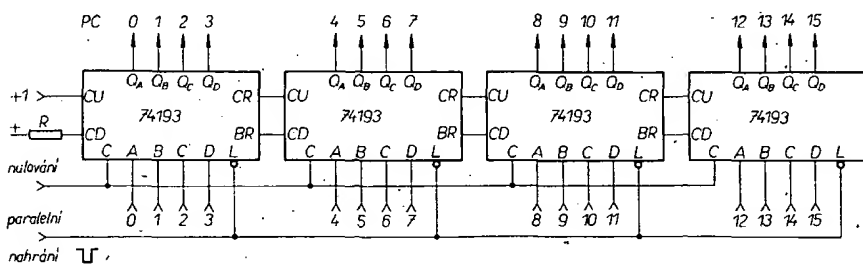
systému. Mikroprocesory vznikly proto, aby nahradily „pevně zapojenou“ logiku v řídicích systémech, terminálech apod. V prvních reklamách firmy INTEL se psalo, že obvod 4004 nahradí asi 40 obvodů MSI TTL a obvod 8008 jejich dvojnásobný počet. V systémech používajících obvody TTL se rozhodování realizuje pomocí hradel, klapných obvodů nebo dekodérů. Chceme-li podobného rozhodování dosáhnout v počítači, musíme to udělat programem. Na obr. 7 je několik příkladů náhrady logických obvodů programem. Počítač se musí umět rozhodnout (a to podle výsledků operace), zda bude pokračovat ve vykonávání další instrukce (výstup AND z rozhodovacího bloku vývojového diagramu), nebo zda přeskočí několik instrukcí. Pak bude realizovat tu část programu, která následuje po výstupu NE z rozhodovacího bloku vývojového diagramu. Podíváme-li se na vývojové diagramy programu, pak vidíme, že se skládají z obdélníků, které představují realizaci nějaké operace (tyto operace provádí v počítači aritmeticko-logická jednotka nebo obvody vstupu a výstupu, které spojují počítač s okolím); kosočtverce ve vývojovém diagramu představují rozhodovací bloky, ty se v počítači provedou tak, že se vlastně změní obsah čítače programu). Právě proto je nutno přikládat tak velký význam pochopení funkce čítače programu. Operace uváděné ve vývojovém diagramu v obdélnících se vlastně moc neliší, ať se již realizují v počítači nebo pevně zapojenou logikou. Způsob rozhodování je však u počítačů naprosto rozdílný. Pevně zapojená logika se může

najednou rozhodnout třeba na základě současné změny čtyř vstupních signálů, prostě se přepne určité klapné obvody, čímž se zablokuje určité cesty a otevrou nové. Počítač však může v jednom okamžiku rozhodnout jen tak, že buď zvětší obsah čítače programu o jedničku nebo změní jeho obsah signálem LOAD. Existují tudíž v jednom okamžiku pouze dvě možné cesty, buď dále, nebo někam jinam.

To, že jsou mikropočítače schopny nahradit logiku, která se může libovolně rozhodovat, je dáno velkou rychlostí, kterou se jednotlivé instrukce realizují. Zkušenosti programátorů mohou nyní namítnout, že i počítač může rozvést cesty při realizaci programu na více směrů než dva. Je to sice pravda, počítač může vlastně naplnit obsah čítače programu výsledkem operace a tím „skočit“ podle něj třeba na 256 míst v programu – to je právě ta výjimka, která potvrzuje pravidlo a vyžaduje znalost funkce čítače programu. Na druhé straně je však nutné si uvědomit, že v tomto případě musí mít počítač všechny signály v jednom výsledku operace (třeba vstupní operace). Změní-li se však čtyři výstupní signály připojené na různé vstupy mikropočítače, musí stejný program realizovat čtyři vstupní operace, výsledek složit a pak se teprve rozhodnout. Dá se tedy říci, že počítač pracuje sice paralelně, ale rozhoduje se sériově, lze říci sekvencně.

Význam funkce čítače programu je ještě zvýšen tím, že mohou být data, která se zpracovávají, a instrukce, podle nichž se data zpracovávají, uloženy na libovolných adresách v paměti počítače. O tom, zda se jedná o data nebo o instrukce, rozhoduje obsah čítače programu. Ten počítá s tím, že na nulté adrese je instrukce a pak je již nutné, aby byl program správně sestaven. Je-li v programu chyba, která způsobí skok na adresu, kde jsou data, interpretuje počítač data jako instrukci a tím dojde nejen k chybnému výsledku, ale program „zabloudí“, a počítač by mohl realizovat operace výstupu, nebezpečné i pro obsluhu řízeného systému, třeba obráběcího stroje.

Máme-li třeba 10 instrukcí za sebou a pak chceme mít data, musí být desátá instrukce skoková, která přeskočí adresu, na níž jsou data. Skoková instrukce je



Obr. 6. Náhradní zapojení čítače programu

vlastně příkaz pro paralelní náhradní nové adresy do čítače programu (signál LOAD). Teď už umíme:

1. Vynulovat čítač programu po zapnutí počítače.
2. Zvětšit obsah čítače programu o jedničku po provedení instrukce, vykonávající nějakou operaci.
3. Buď o jedničku zvětšit nebo zcela změnit obsah čítače programu na základě výsledku nějaké operace. Tomuto případu říkáme podmíněný skok.
4. Instrukci skoku paralelně naplnit čítač programu novou adresou. Tomuto případu říkáme nepodmíněný skok.

Tím však nejsou vyčerpána všechna kouzla, která dělá počítač s obsahem programového čítače. Má-li program mikropočítače ušetřit kapacitu poměrně drahých pamětí EPROM, musí být co nejkratší. Jednou z cest, jak dosáhnout krátkého programu, je používat podprogramy. Podprogram je obvykle krátký program, který je využíván programátorem k plnění často se opakujících úloh. Podprogramem může být třeba násobení nebo tisk zadaného počtu znaků na tiskárně. Místo toho, abychom kdykoli, potřebujeme-li vytisknout znak, napsali instrukci výstupu, napíšeme raději skok do podprogramu, který zajistí komunikaci s tiskárnou. Jak jsme si již řekli, skok je vlastně paralelní naplnění čítače programem, v tomto případě se čítač naplní adresou, na které je první instrukce podprogramu tisku. Program, který využívá podprogramu, si vlastně šetří místo v paměti a po realizaci tisku pokračuje další instrukcí, která následuje ihned za skokem do podprogramu. Počítač proto nemůže pouze měnit obsah čítače programu, ale musí si nejprve zapamatovat jeho starý obsah, aby věděl, kam se má z podprogramu vrátit. Vyplývá z toho další operace s čítačem programu, tj. uložení jeho starého obsahu a jeho obnovení při návratu z podprogramu. Starý obsah se ovšem musí obnovit již zvětšený o jedničku, aby se program vrátil za instrukci skoku a ne znovu na ni.

Opět to však není vše, co počítač dokáže manipulaci s obsahem čítače programu. Jak uvidíte sami, až budete programovat, počítač komunikuje s přídatnými zařízeními třemi způsoby: programově, nebo přerušovacím systémem počítače, nebo metodou přímého přístupu do paměti. Hlavním problémem však není způsob přenosu dat mezi počítačem a přídatným zařízením, ale způsob synchronizace počítače a přídatného zařízení. Potřebuje-li počítač přečíst všechna data z děrné pásky, musí nejprve zjistit, jsou-li data na vstupech ze snímáče platná, tzn. že se nemění. Kdyby počítač četl data podle svého vlastního vnitřního kmitočtu, v jehož rytmu se vykonávají instrukce, často by se stalo, že by přečetl stav signálů např. z fototranzistorů v okamžiku, kdy se mění, nebo by stačil přečíst již jednou převzatý znak vícekrát. U rychlého zařízení, třeba magnetické páskové paměti, by se zase mohlo stát, že by počítač přebíral data příliš pomalu a některé znaky by nepřečetl. Synchronizace vstupu a výstupu je proto naprosto nezbytná a tvoří velkou část problematiky, které říkáme připojování vstupních a výstupních zařízení, neboli INTERFACING.

Čítač programu se používá při synchronizaci počítače s okolím pomocí přerušování. Přerušování je externí signál, který, je-li počítačem akceptován, vnutí za právě ukončenou instrukci navíc instrukci skoku. Jde vlastně o naplnění čítače programem buď pevně stanoveným obsahem nebo daty, která jsou přijata z obvodů vstupů a výstupů. Data, která se použijí

pro tento účel, nazýváme vektorem přerušování. Vektor pošle zařízení, jehož přerušování bylo akceptováno, vlastně adresu první instrukce obsluhujícího podprogramu pro toto přídatné zařízení.

Je samozřejmé, že platí to, co platilo obecně o podprogramech. Ušchová se starý obsah čítače programu a po skončení práce podprogramu se program vrátí na adresu následující za instrukcí, která způsobila přerušování programu.

Jak je vidět, čítač programu je skutečně základem počítače a také klíčem k pochoopení činnosti počítače. Naproti tomu aritmeticko-logická jednotka, pracovní registry, registr a dekodér instrukcí jen připravují pro obvody řadiče příznaky, podle nichž se pak ovládají řídicí signály pro čítač programu. Programujeme-li počítač, je vlastně činnost čítače programu víceméně skryta za symbolické názvy adres skoků a podprogramů. Chceme-li však mikropočítače skutečně využít i pro rychlé aplikace, musíme si umět představit, jak pracuje. Jak lze takového mikropočítače využít, je názorně vidět na osobním počítači ZX80, u něhož je právě čítač programu základem celého „triku“, využitého ke kreslení znaků na obrazovku s minimem součástek.

Mohl bych takto pokračovat ještě dlouho a popisovat, jak vlastně počítač pracuje. O práci počítačů je však poměrně dost literatury a tak ať tyto řádky pomohou těm, kteří se v problematice dosud neorientovali.

Co potřebujeme k práci s mikropočítači?

Koupíme-li si auto, dostaneme k němu velmi chudou literaturu o údržbě a použití. Přesto se však většina z nás brzy dopracuje k tomu, že si umí leccos vyměnit a opravit. Něco odkoukáme od zkušenějších, něco získáme z literatury a něco si zkusíme – i když se nám to třeba hned nepovede, příště to jde lépe. S mikropočítačem je to obdobné. Stejně jako u auta, kde také nezačínáme s generálkou motoru, musíme postupovat od jednoduchých věcí k složitějším. Nejlepší je, když se zpočátku vyhneme problematice vlastního procesoru a převezmeme již osvědčené zapojení a základní programy. Bylo by ideální, kdybychom si osobní mikropočítač mohli koupit a zabývat se připojováním přídatných zařízení a psaním aplikačních programů. Vlastní práci s mikropočítačem získáme pak více vědomostí, než pouhým čtením literatury, která je často odtržená od praxe.

Vědomosti pro práci s mikropočítačem není možné získat najednou tak rychle. Je třeba prostudovat základní literaturu a pak si něco zkusit postavit a přeměřit, zda se „to“ chová tak, jak jsme předpokládali. Já sám jsem si vzal přepínače, diody LED a odpory a tak jsem si ověřil funkci obvodu MHB3212. I když se zdá, po přečtení katalogového listu tohoto obvodu, že je vše naprosto jasné, není to většinou pravda. A to je obvod MHB3212 jedním z nejjednodušších obvodů, používaných pro stavbu mikropočítačů. Přestože s těmito obvody pracuji již rok, musím se přesto někdy znovu podívat do sešitu na funkci jednotlivých signálů.

Základem studia mikropočítačů je nutnost získat alespoň základní literaturu. Není to vždy jednoduché. Obvod MHB1012 je vyráběn již dva roky, aniž by o něm byly publikovány základní informace. Jediné díky tomu, že je ekvivalentní nejjednoduššímu obvodu UART na světě, AY-5-1013 (TMS6011), je vůbec možné s ním pracovat. Proto je důležité před-

kreslovat si do sešitu různá zapojení obvodů ze zahraniční literatury a získávat kopie dokumentací, katalogů, knížek. Důležité je odhadnout, co budeme skutečně potřebovat a nevytvářet stohy materiálů o obvodech, s nimiž budeme těžko někdy pracovat. Ze svých zkušeností mohu doporučit články z roku 1976–78, kdy se v zahraničí nejvíce psalo o aplikacích mikropočítačů 8080A. Nesmíme zapomenout na to, že obvody TTL řady 74 a obvody CMOS řady 4000 jsou pro mikropočítače stejně důležité jako procesory nebo paměti. Proto je nutné nadále sbírat a studovat zapojení těchto obvodů, neboť na nich se pak dá nejvíce ušetřit při návrhu mikropočítače.

Součástková základna pro stavbu mikropočítačů je u nás velice chudá. V tab. 1 je přehled součástek pro osmibitové mikropočítače, v tab. 2 pro šestnáctibitové mikropočítače a v tab. 3 jsou paměti. Tyto součástky jsou zařazeny do „Perspektivní řady součástek pro elektroniku“ a v tabulce je uveden i plánovaný rok výroby. Pro stavbu mikropočítače s procesorem 8080A budeme mít téměř všechno. Bohužel však chybí běžné obvody TTL a obvody řady 74LS, které jsou pro práci s mikropočítačem velice nutné. Každý postupně narazí na problémy, které jsou bez obvodů používaných v zahraničních zařízeních jen těžko řešitelné. Chybí nám 8bitové

Tab. 1. Obvody 8bitových mikropočítačů

| Typ | Funkce | Pozn. |
|----------|---------------------------|------------|
| MHB1012 | UART | vyrábí se |
| MHB8080A | mikroprocesor | 1982 |
| MHB8251 | UART/USART | 1983 |
| MHB8255 | paralelní vstup/výstup | 1983 |
| 8253 | časovač | 1981, SSSR |
| 8257 | řadič DMA | 1981, SSSR |
| 8259 | řadič přerušování | 1981, SSSR |
| 8048 | jednotčipový mikropočítač | 1985 |
| 8035 | jednotčipový mikropočítač | 1985 |
| 8243 | vstupy a výstupy pro 8048 | 1985 |
| MH8224 | hodinový obvod 8080A | 1983 |
| MH8228 | systémový obvod 8080A | 1983 |
| MH3205 | dekodér adres | vyrábí se |
| MH3212 | 8bitový port | vyrábí se |
| MH3214 | obvod přerušování | vyrábí se |
| MH3216 | 4bitový budič | vyrábí se |
| MH3226 | 4bitový budič | vyrábí se |

Tab. 2. Obvody 16bitových mikropočítačů

| Typ | Funkce | Pozn. |
|---------|------------------------|------------|
| 8086 | 16bitový mikroprocesor | 1983, SSSR |
| MHB8282 | 8bitový střadač | 1984 |
| MHB8283 | 8bitový střadač | 1984 |
| MHB8284 | hodinový obvod | 1984 |
| MHB8286 | 8bitový budič | 1984 |
| MHB8287 | 8bitový budič | 1984 |

Tab. 3. Paměti pro mikropočítače

| Typ | Funkce | Pozn. |
|---------|-----------------------|-----------|
| MHB1902 | RAM statická 1K1 CMOS | vyrábí se |
| MHB2102 | RAM statická 1K1 | vyrábí se |
| MHB4116 | RAM dynamická 16K1 | vyrábí se |
| MHB2114 | RAM statická 1K4 | 1984 |
| MHB8111 | RAM statická 256 x 4 | 1983 |
| MHB2708 | EPROM 1K8 | 1983 |
| MHB2316 | ROM 2K8 | 1983 |

budiče sběrnic (74244), tvarovací obvody 7414 nebo 74132, hradla OR 7432, komparátor 7485, registry 74175 a 74174 a hlavně alespoň základní třístavové hradlo typu 74126. Proto je někdy nutné použít i obvody ze zahraničí. To si však může dovolit amatér a ne vývojový pracovník, který připravuje zařízení pro sériovou výrobu. Proto jsou naše zařízení dost složitá a mají velké nároky na výkon napájecích zdrojů. Ono použít místo hradla 74126 obvody 3212 nebo 3216 je pro konstruktéra vlastně přestupek proti snaze, dělat zařízení dobrá, spolehlivá a levná. Co má však dělat, potřebuje-li přenést třeba jeden stavový bit z tiskárny na sběrnici?

Dále je třeba pro práci s mikropočítačem mnoho konektorů, objímek pro integrované obvody, přepínačů, tlačítek a kabelů. Zejména nedostatek levných a dobrých konstrukčních součástek způsobí, že mikropočítače budou pro řadu amatérů ještě dlouho nedostupné. Některé z těchto součástek je možno i nahradit, ale nesmí to být na úkor spolehlivosti, protože mikropočítač s jediným nespolehlivým kontaktem je naprosto nepoužitelný.

Klíčovým problémem amatérské stavby mikropočítače budou desky s plošnými spoji. Oboustranné desky s plošnými spoji s prokovenými děrami jsou u mikropočítačů běžné a amatér si je zhotovit nemůže. Dá se sice dělat i bez nich, ale pak je určitě lepší „zadrátovat“ zapojení na univerzální desce s plošnými spoji. Jedním z úkolů tohoto čísla AR je ukázat na problémy, které vzniknou při stavbě mikropočítačů. Rozdíl mezi systémem INTELKA a JPR-1 je vlastně pouze v použitém konektoru a technologii přípravy pro desky s plošnými spoji a v technologii jejich výroby.

Máme-li dostatek zkušeností, literatury, součástek a možnost si navrhnout a zhotovit desku s plošnými spoji, můžeme se pustit do konstrukce mikropočítače.

Nejvyšší nároky kladé stavba mikropočítače právě na konstrukční práci. Vše je třeba dělat velice pečlivě. Počítače, a nakonec i programy pro ně, mají tu vlastnost, že jsou-li dobře postaveny, pracují, jak se říká, na první zapnutí. Běda však, je-li někde chyba. Pak se můžeme obklopit logickými analyzátory a osciloskopy a chybu budeme stejně těžko hledat. U analogové techniky byla vždy možnost, že chyba způsobila jen špatnou funkci zařízení. I u číslicové techniky pracovala alespoň část zařízení dobře a chybu bylo možno nalézt sondou nebo osciloskopem. U počítače způsobí obvykle i malá chyba takovou změnu v chování systému, že nevíme, kde máme s hledáním začít. U mikropočítačů je také velké množství konektorů a kabelů a stačí přehodit dva

vodiče a celý systém se zablokuje, třeba proto, že čeká na stlačení tlačítka, jehož stav se přenáší po chybném spoji. Proto je pro mne nejpoužívanějším nástrojem při stavbě mikropočítače mikrotužka. Jeden výkres má dnes několik stovek čísel a čar. Zkontrolovat návrh plošných spojů trvá deset hodin a překreslení návrhu několikrát déle. Je samozřejmé, že se nevyhne chybám, ale musíme se snažit, abychom svou pečlivostí zmenšili jejich počet na minimum. Pro konstrukční práce potřebujeme mít dostatek podkladů. Musíme mít zapojení vývodů jednotlivých obvodů, jejich parametry, časové diagramy signálů, normy pro kreslení plošných spojů a katalogy pasivních a konstrukčních součástek. Chceme-li udělat mikropočítačový systém malý a kompaktní, nemůžeme oddělovat vlastní vývoj zapojení od konstrukční práce. Někdy se nám stane, že až při návrhu plošných spojů musíme předitel zapojení, aby někdy nezbylo příliš mnoho nevyužitých hradel v obvodu, nebo musíme přečíslovat vývody konektorů. Při konstrukci systému nesmíme zapomínat na to, že kvalita mechanické konstrukce přímo určuje jeho spolehlivost. Většina poruch počítačů je způsobena namáháním desky nebo konektorů při jejich zasouvání, namáháním vodičů v kabelech nebo špatným chlazením součástek.

Desky mikropočítačového systému jsou mnohem složitější, než tomu třeba bylo u čítače nebo digitálního voltmetru. Proto doporučuji zapojení desky vždy nejprve ověřit.

Nejlepší technologie pro zapojení univerzálních desek – (já jim říkám basti desky) – byla popsána v Ročence AR 1981. Tuto technologii je možno použít i pro stavbu celého mikropočítače. Jednotlivé desky se propojují měděným vodičem se speciální izolací, která se sama při pájení odstraní. Bohužel, ověření desky vyžaduje objímky pro všechny integrované obvody.

Řekněme si teď, co vše potřebujeme pro měření a ožiování desek mikropočítačového systému. Nejčastější závadou na deskách jsou zkratky a chybějící nebo přehozené spoje. Základními přístroji proto nadále zůstávají:

1. Ohmmetr s rozsahem 10 Ω až 10 k Ω pro měření zkratů, přechodů polovodičů, odporů a pro zkoušení LED. Zkratový proud ohmmetru by měl být asi 5 mA.
2. Voltmetr s rozsahem ± 15 V k měření napětí. Doporučuji změnit na desce všechny vývody všech obvodů, zda mají správné napájecí napětí. Nezapojený vývod země se jinak hledá velmi špatně. Rovněž je nutné měřit napětí na všech kontaktech objímek, než do nich zasuneme drahé integrované obvody.
3. Logická sonda, která je schopna změnit tři stavy.
4. Miliampérmetr s rozsahem asi 10 mA pro měření zatěžovacích proudů. Obvody MOS mají výstupní zesilovače navrženy na proudy asi 2 mA. Je lépe se přesvědčit přímo na objímce nebo konektoru před zasunutím obvodu nebo desky, zda nebude proud větší při chybě v zapojení nebo zkratu na desce s plošnými spoji.
5. Napájecí zdroj +5 V/3 až 5 A, +12 V/1 A, -5 V/0,5 A a -12 V/0,5 A. Zdroj by měl mít měřidlo odebraného proudu a musí být spolehlivý. Zejména nesmí při zapínání a vypínání produkovat přepětové špičky! Je lepší mít napětí vyvedena přes konektor, který nelze otočit, než přes obvyklé zdířky. V zapojování vodičů do zdířek se často udělá chyba.

6. Nářadí se používá stejně jako pro práci s obvody TTL: páječka, skalpel, pinseta, štipačky (jedny se zabroušenou čelistí na přeštipávání vývodů vadného obvodu), lupa a kartáček na čištění desky. Je třeba mít nějaký nástroj na „vydloubnutí“ obvodů z objímky. Před zasunutím obvodu do objímky musíme srovnat, zejména u nových obvodů, vývody do správné rozteče. Dělá se to tlakem o desku stolu. Páječka nemusí být speciální. I když by byla někdy lepší mikropáječka, používám pistolovou páječku, protože dobře prohřeje izolaci samopájitelného vodiče. Pájím-li na bodu, který je přímo spojen s drahým obvodem, vyjmu tento obvod z objímky. Nepoužívám žádný systém zemnění při manipulaci s obvody, pouze se nejprve dotknu spoje desky nebo vodiče podložky, do které se obvody dávají, jsou-li mimo zařízení. To však neznamená, že nejsem opatrný při práci, vím, co obvod sнесou a co ne. Spěch při práci s obvody MOS a CMOS se rozhodně nevyplácí!

7. Velkým problémem je vrtání desek. Při malých průměrech pájecích děr je třeba používat stojanovou vrtačku a vidiové vrtáky o \varnothing 0,6 až 0,8 mm a odpovídající rychlosti otáčení. Nejsou-li díry prokovené, propájím součástku shora i zespodu. Tam, kde bude objímka, je nutné prostrčit tenký pocínovaný drátek a připájet ho nahore po délce spoje asi 0,5 cm. Až jsou takto připraveny všechny vývody obvodu, které procházejí dolů, zastrčíme objímku a zapájím vývody objímky i konce drátků dole.

8. Přístrojové vybavení není třeba speciální, kromě osciloskopu (alespoň 10 MHz). Osciloskop by měl být dvoukanalový. Pro indikaci průběhů na několika místech najednou lze pak udělat elektrický přepínač třeba na 16 kanálů. Ten se však k měření příliš nehodí. Generátor impulsů, čítač a nakonec i jednoduchý logický analyzátor si také můžeme zhotovit sami.

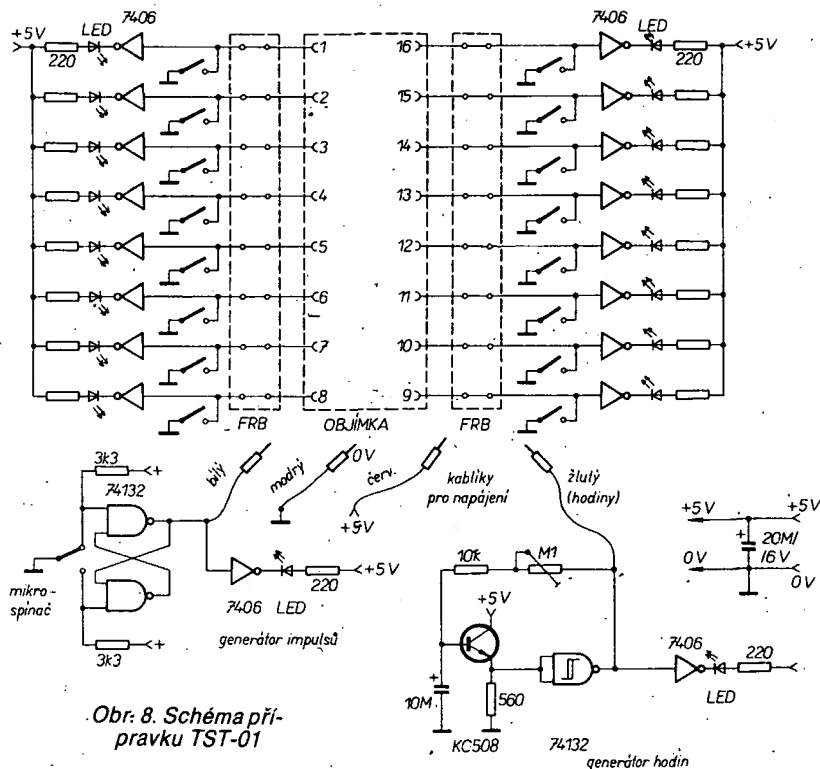
Jak poznáte sami, až budete s mikropočítačem pracovat, není někdy jednoduché přijít na to, jak vlastně postupovat dál. Při ožiování nebo hledání závad v mikropočítačovém systému se vyplatí raději hned nic konkrétního nedělat, když nevíme co, mnohem lépe je přemýšlet, jak postupovat. Je dobré, pokud možno, daný problém co nejvíce zjednodušit, nebo „izolovat“ od okolí. To znamená, že když je podezření na paměť, je vhodné najít cestu, jak ji vyzkoušet bez procesoru; je-li podezření na procesor, je vhodné ho vyzkoušet bez paměti.

U logických systémů s obvody TTL jsem si nejčastěji pomáhal pravidlem, že jeden výstup obvodu z jednoho pouzdra se smí uzemnit. Vzal jsem do jedné ruky jehlu připájenou k uzemněnému lanku a do druhé ruky sondu a postupně jsem měřil chování hradel a klopných obvodů. Tento postup u obvodů MOS nedoporučuji! Chceme-li použít metodu uzemňování, musíme se nejprve přesvědčit, zda jsou všechny výstupy obvodů připojených do bodu, který chceme zemnit, ve třetím stavu. Většina třístavových obvodů v mikropočítačových systémech má povolovací vstupy (CE, CS, OE atd.) ovládané na základě dekódování adresy, případně součinem platné adresy a řídicího signálu (WE, MR, atd.). Proto je nejprve nutné prověřit obvody adresace, potom funkce řídicích signálů a nakonec data. Při tomto postupu začínáme s ožiováním se všemi obvody vyjmutými z objímek. Vlastní postup ožiování desek mikropočítačového systému je popsán u jednotlivých dílů systému JPR-1.

Dále si ukážeme konstrukci jednoduchých, ale velmi užitečných pomůcek při práci s mikroprocesory a mikropočítači.

Tester TST-01

Tento tester vlastně ani do mikropočítačové techniky nepatří. Je totiž určen pro zkoušení obvodů TTL. Obvod TTL je špatný asi tak jeden ze sta, u některých typů a sérií i pět ze sta. Obvody TTL předem obvykle neměřím. V mikropočítačovém systému je však lepší změřit obvody předem, alespoň po funkční stránce, neboť výměna vadného IO má obvykle za následek zničení části drahé desky s plošnými spoji. Na obr. 8 je schéma TST-01. Podobný přípravek mám pro měření obvodů s počtem vývodů 28 a 24, zejména pro obvody mikroprocesorové řady 3000. TST-01 má objímku pro měřený obvod, 16 indikátorů úrovní na jednotlivých vývodech, 16 přepínačů pro vnučení úrovně log. 0 a generátor impulsů. Volba napájení je umožněna kabely „0 V“ a „+5 V“. Dále je na TST-01 jeden mikrospínač, ošetřený klopným obvodem R-S, indikací stavu jeho výstupu. Vývody pouzdra jsou vyvedeny na „kousky“ konektoru FRB, aby bylo možné propojit jednotlivé vývody navzájem kabelky se špičkami FRB a připojit napájení, generátor nebo klopný obvod R-S. Před zasunutím obvodu je třeba zkontrolovat správné napájení a stav přepínačů, aby nebyl zkratován některý z výstupů. Postup měření vymyslíte z katalogu a alespoň lépe poznáte funkci obvodu.



Obr. 8. Schéma přípravku TST-01

prázdných objímek pro obvody 8080A a 8224, získáme „tlačítkový mikroprocesor“. Jeho pomocí zkontrolujeme adresaci portů a paměti a zkontrolujeme signály datové, adresové a části řídicí sběrnice. Máme-li pak hotový systém JPR-1, může-

me pomocí TST-03 a kabelu KB-03 krokovat programy, čist obsah paměti nebo jej měnit.

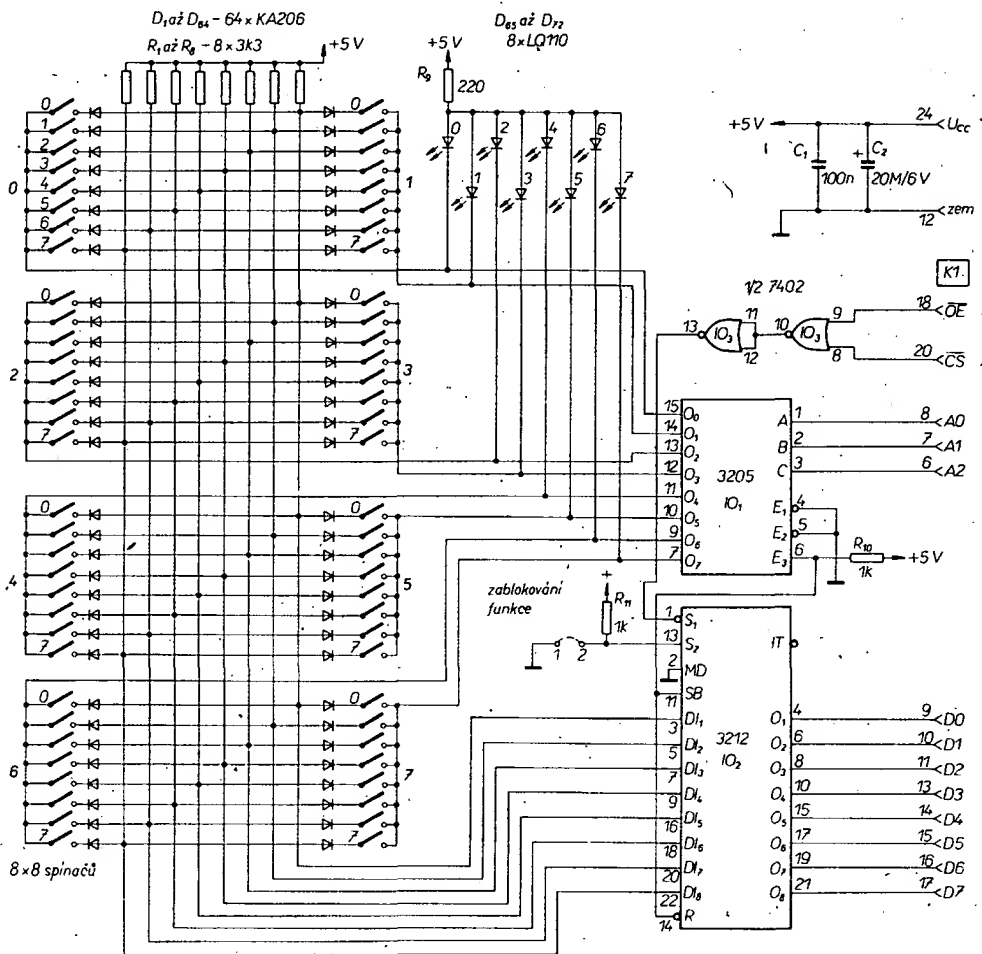
Schéma přípravku TST-03 je na obr. 11, jsou v něm použity obvody 74125 (je možno použít s malými změnami 74126),

Přípravek TST-02

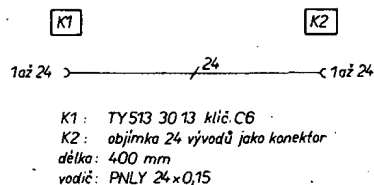
Tento přípravek je vlastně paměť EPROM s kapacitou 8 bytů. Kdybychom chtěli nahradit celou paměť 2716 a naskládati stejné přípravy na sebe, měl by celek výšku asi 5 metrů. Při ožiování desky procesoru nebo desky paměti nám však malá kapacita paměti zcela vyhoví. Při ožiování procesoru postačí provádět jednu instrukci (třeba „skok na sebe“) nebo jednu instrukci a skok zpátky. Přípravek TST-02 je vlastně diodová paměť PROM, u níž jsou přepalovací můstky nahrazeny přepínači. To, že jsou v mém přípravku použity přepínače DIL, nesmí nikoho odradit. Dlouho jsem používal stejný přípravek s přepínači Isostat a podobný se síťovými spínači. Schéma přípravku TST-02 je na obr. 9. Přípravek se připojuje kabelem (kabel KB-04, obr. 10) do objímky obvodů paměti EPROM 2708 nebo 2716. Z těchto důvodů jsou signály ze špiček CS a OE sečteny hradlem 7402. Celou „paměť“ je možno zablokovat spojením spojky 1-2. U každého byte svítí dioda LED, aby bylo možno při krokování programu sledovat adresaci. Vzhledem ke skladnosti přípravku je kabel vyveden přes konektor FRB o 30 vývodech. Pak lze stejný kabel použít pro připojení simulátoru EPROM. Napájecí napětí přípravku TST-02 se odebírá z objímky paměti EPROM.

Přípravek TST-03

Tento přípravek je již trochu složitější, má však i značně široké použití. Vývody přípravku jsou připojeny na konektor, zapojený stejně jako konektory sběrnice ARB-1 systému JPR-1. Pomocí TST-03 je tedy možno ožiovat desku procesoru, sběrnici a celý systém JPR-1. Připojíme-li ke konektoru K1 přípravku TST-03 kabel KB-02 a jeho druhý konec zasuneme do



Obr. 9. Schéma přípravku TST-02



Pozn. 1 – spojit špičky se stejnými čísly
Pozn. 2 – přes pájené špičky K1 navléknout bužírku
Pozn. 3 – do objímek pájet seshora, rychle, a pájenou objímku zasunout do pomocné objímky, aby se vývody při ohřátí nehnuly

Obr. 10. Kabel KB-04

protože tyto obvody jsou v naší perspektivě součástkové základně. Bez těchto obvodů se prostě s mikropočítači nemá vůbec dělat! Je sice možno použít i obvody 7403 s otevřenými kolektory, ale třístavové obvody by měly spolupracovat s obvody, které mají zátěž stejných vlastností. Přesto vám doporučuji přípravek si postavit alespoň s obvody 7403. Na obr. 14 jsou

potřebné změny v zapojení přípravku vyznačeny. Práce s přípravkem TST-03 je popsána u ožiování desek JPR-1, AND-1 a REM-1. Sběrnici ARB-1 doporučuji oživit bez přípravků, její špatné zapojení může zničit i přípravek TST-03. Pro hledání závad na „porouchané“ sběrnici je však TST-03 vhodný. Na obr. 12 je zapojení kabelu KB-02 a na obr. 13 zapojení kabelu KB-03.

Simulátor EPROM

Simulátor EPROM je vlastně paměť RAM, pracující místo paměti EPROM. Je to nejužitečnější přípravek pro mikropočítače. Snad právě proto se u nás nevyrábí. Místo něj se používají emulátory procesorů ve spolupráci s vývojovými systémy. Pro ty, kteří mikropočítačům skutečně rozumějí, je emulátor pomůckou. Pro první práce s mikropočítači a pro ladění programů „v poli“, to znamená na místě, na němž je mikropočítač nasazen, simulátor EPROM zcela vyhoví.

Zatím se seznámíme s blokovým schématem simulátoru EPROM. Mám sice jeden simulátor v provozu, ale jeho popis a konstrukce si zaslouží samostatný článek.

Simulátor EPROM pracuje podle blokového schématu na obr. 15. Jeho základní funkce jsou:

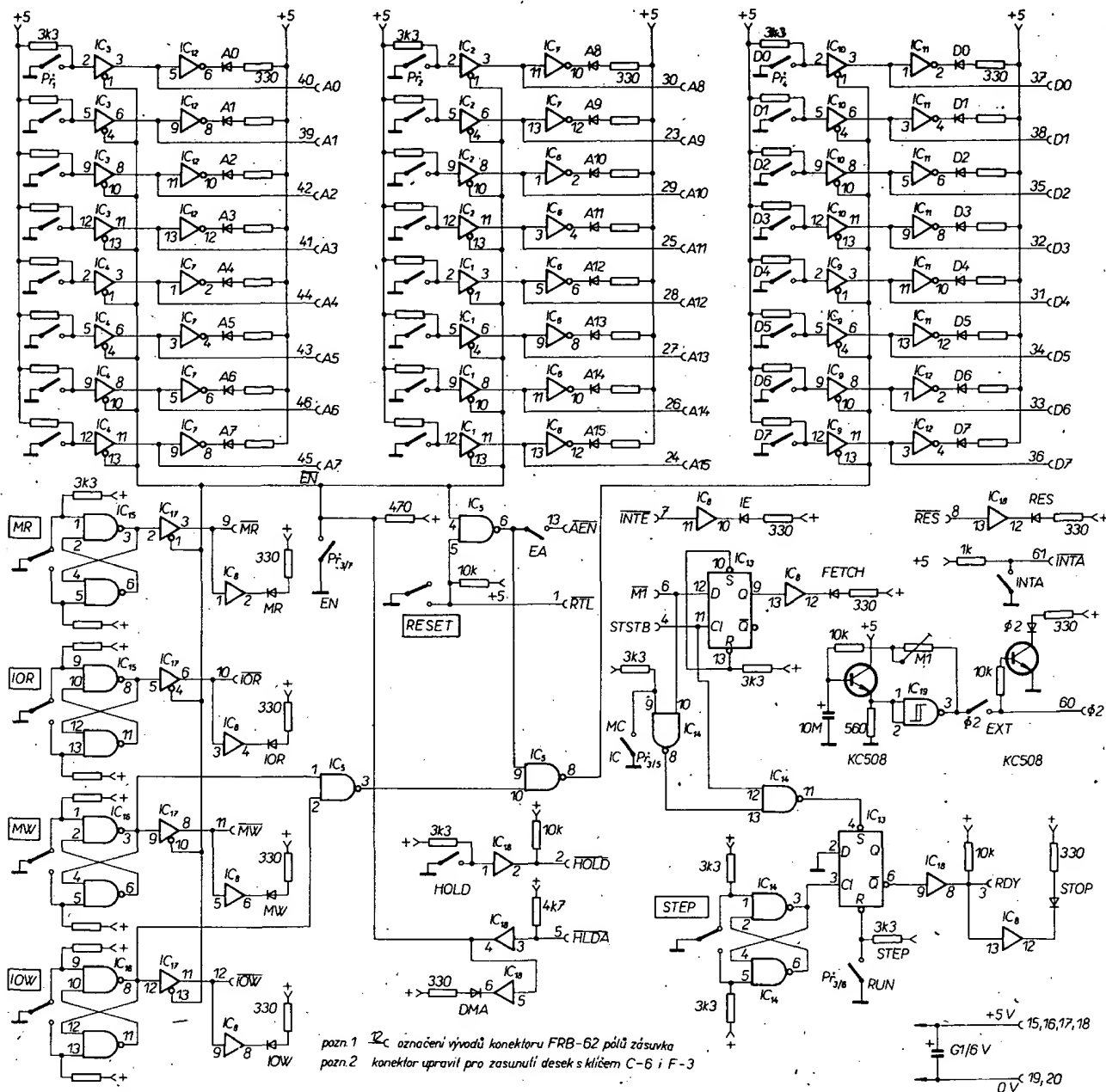
0. Programování – obsah vnitřní paměti simulátoru RAM se přehraje do paměti EPROM 2708 nebo 2716.

1. Přenos (TRANSFER) – obsah paměti EPROM zasunuté do objímky na simulátoru se přenesou do vnitřní paměti RAM simulátoru.

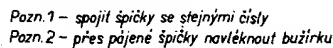
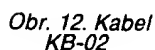
2. Vstup (INPUT) – procesor může „přehrát“ libovolná data do paměti RAM simulátoru. Použije k tomu kabel, pomocí něhož je simulátor připojen do objímky některé paměti EPROM mikropočítačového systému.

3. SIMULACE – vnitřní paměť simulátoru RAM, se chová jako paměť EPROM, do jejíž objímky je simulátor připojen.

4. Běžný běh programu (RUN) – paměť EPROM zasunutá do objímky na simulá-



Obr. 11. Testovací přípravek TST-03



Obr. 13. Kabel KB-03



Využití simulátoru EPROM je značně široké. Máme-li simulátor, šetříme tím paměti EPROM, neboť je nemusíme mazat po chybném pokusu o napsání programu. Simulátor je výhodný zejména proto, že se do malého přístroje soustředí téměř všechny operace, které musíme dělat při ladění a opravách programu. Ve

Je samozřejmé, že pro práci s mikroprocesory potřebujeme mnohem víc, než jsem uvedl. Potřebujeme mazat paměti EPROM světlem UV, měřit dynamické parametry některých součástí, analyzovat průběhy signálů a další a další věci. To, co jsem uvedl, je však ukázkou základního přístupu k celé problematice a ten spočívá v tom, že se nejsložitější věci musí řešit co nejjednodušeji, jinak je nezvládneme. Vybavení, znalosti a zkušenosti rostou podle „lineární a spojitě“ funkce a někde musíme začít. Tak hlavně se nebát a směr je dopředu!



6. Kontrola (VERIFIKACE) – obsah paměti EPROM zasunuté do objímky na simulátoru se porovná s obsahem vnitřní paměti RAM simulátoru. Najde-li se rozdíl, indikuje se adresa a data.

II. Amatérský osobní mikropočítač INTELKA

Jaromír Šíma, Michal Humpál

Intelka je osobní amatérský mikropočítač s mikroprocesorem INTEL 8080A. Může to být samozřejmě mikroprocesor jiného výrobce (např. TESLA MHB8080A). Písmeno A za označením mikroprocesoru vyjadřuje, že signály pro přímý přístup do paměti (DMA), signály pro přerušení a vstup připraven (READY) jsou již uvnitř mikroprocesoru zasynchronizovány s hodinami Φ_2 . V současné době se jiný druh μP 8080 nevyrobí.

Celý mikropočítač je vestavěn do konstrukce bulharské stolní tranzistorové kalkulačky ELKA. Spojením názvu INTEL a ELKA vznikl název tohoto mikropočítače. Při konstrukci a návrhu Intelky bylo sňahou splnit několik požadavků, které si však navzájem tak trochu odporují: jednak získat co možno nejuniversálnější systém, který by si mohl každý rozšířit podle svých představ, jednak zhotovit systém co nejjednodušší, používat nejdostupnější součástky a přitom nezapomenout na určitou „bytnost“ systému a odolnost proti všetečkům.

Jako každý mikropočítač se Intelka skládá z několika nutných celků: ze zdroje, sběrnice, centrální jednotky CPU, operační paměti a vstupních a výstupních obvodů (porty).

Popis zapojení

Zdroj

K napájení Intelky potřebujeme několik stabilizovaných napětí: +5 V (nejméně 5 A), +12 V a -12 V (asi 0,5 až 1 A). (Počítá se i s napájením operačních zesilovačů pro převodníky apod.). Zdroj lze realizovat např. podle návodů v ARB č. 4/82, nebo použít zdroj pro napájení JPR-1. Vlastní zdroj pro Intelku, který je vyřešen konstrukčně pro vestavění do vany, bude uveřejněn v některém z dalších čísel AR.

Sběrnice

Amatérská výroba vany (celku s konektory a vodičmi lištami) je náročná a konektory jsou drahé. Proto byla použita mechanika z dostupné bulharské kalkulačky ELKA. Tato kalkulačka se vyřazuje z podniků a závodů a je přístupná amatérům. Zapojení sběrnice je na obr. 1. Signály sběrnice se dělí na napájení, adresové signály, datové signály, řídicí signály a signály pro přímý přístup do paměti (DMA). Všechny signály na sběrnici (kromě datových signálů) jsou vzhledem k volbě obvodů 7438 (7417) jako nejlepšího budiče sběrnice NEGOVÁNY! Na to musíme dát pozor při konstrukci a návrzích dalších obvodů.

Adresové signály A0 až A15 zahrnují 16 adresových linek pro adresaci operační paměti o 64K byte (8 vyšších a 8 nižších adres). K adresaci vstupního nebo výstupního zařízení (periferie) se používá pouze 8 nižších adresových linek. Proto periférii může být jen 256.

Signál REFRESH oznamuje, že na adresových linkách je přítomna občerstvovací adresa paměti. V tomto systému je

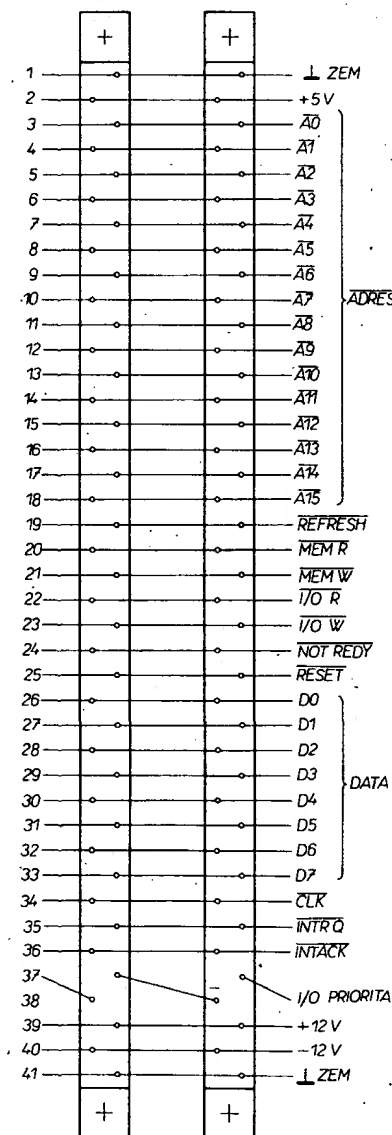
signál nevyužit, je ho možno použít s procesorovou deskou s μP Z80.

Signál MEM R oznamuje paměti, že mikroprocesor chce číst data na adrese, kterou vysílá na adresovou sběrnici. Signál MEM W oznamuje paměti, že mikroprocesor chce uložit data, která vysílá na datovou sběrnici, do paměti, a to na adresu, umístěnou na adresové sběrnici.

Signály I/O R a I/O W jsou podobné signálům pro čtení a zápis do paměti, ale adresa periferního zařízení se vysílá jen na 8 nižších adresových linkách.

Signálem NOT READY oznamuje paměť nebo periferní zařízení, že nejsou schopny vyhovět požadavkům mikroprocesoru (čtení nebo zápis dat). Mikroprocesor přejde do stavu WAIT, kde čeká, až bude jeho požadavek uspokojen. Jeho práce je zastavena a mikroprocesor neodpovídá na požadavky pro přímý přístup do paměti nebo na přerušení.

Výstupní signál RESET oznamuje ostatnímu zařízení, že je mikroprocesor vynulován a začíná pracovat s pamětí od nulové adresy.



Obr. 1. Sběrnice mikropočítače Intelka

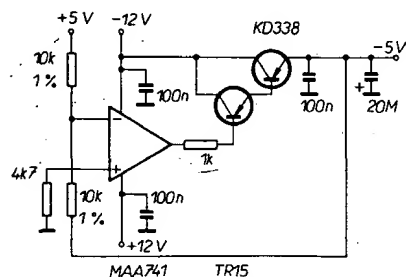
Datová sběrnice D0 až D7 je vedena přímo z obvodu 8228. Její signály jsou proto jako jediné třístavové. Ostatní signály jsou zesíleny obvody s výstupem s otevřeným kolektorem (7438, 7417).

Signál hodin CLK slouží pro synchronizaci všech ostatních částí systému s mikroprocesorovou deskou. Signálem INTRO oznamuje periferní zařízení mikroprocesoru svou žádost o přerušení. Mikroprocesor dokončí instrukci a signálem INTACK potvrdí požadavek na přerušení. V tomto okamžiku periferní zařízení musí vyslat na datovou sběrnici instrukci RST 0 až RST 7, popř. jinou jednoobytovou instrukci (a tím vyvolá skok do obslužného podprogramu pro tuto periférii).

Priority jednotlivých zařízení (tzn., která periférie může vyvolat další přerušení programu při obsluze jiné periférie mikropočítačem) se dosáhne pomocí I/O prioritního řetězce. Toto uspořádání šesti množství špiček sběrnice, má však nevýhodu: priority jednotlivých periférií nelze v průběhu programu měnit. Změnit ji však můžeme změnou pořadí jednotlivých periferních desek v mikropočítači, tudíž ručně a naprosto na delší dobu. Takové konstrukční řešení v běžném provozu vyhoví i náročnějším amatérům.

Signály pro přímý přístup do paměti jsou pro nedostatek špiček sběrnice vedeny k deskám ze zadní sběrnice (na opačné straně desky naproti konektoru sběrnice hlavní). Signálem HLDRQ se požaduje po mikroprocesoru přímý přístup do paměti. Mikroprocesor dokončí operaci se sběrnici a své výstupy uvede do třetího stavu. Současně signálem HLDA potvrdí požadavek na DMA. Mikroprocesor zatím čeká naprázdno na uvolnění sběrnice. DMA se používá pro přesuny velkých celků dat, např. z magnetické pásky (nebo i při obnovování dynamické paměti RAM).

Napájecí napětí pro mikropočítač je vedeno po krajích konektoru hlavní sběrnice. Zem 0 V a napětí +5 V se rozvádějí nejlépe měděným páskovým vodičem na proudové zatížení 20 A – je sice předimenzován, ale z hlediska úbytku napětí se to vyplatí. Napětí +12 V a -12 V rozvedeme měděným vodičem o \varnothing 1 až 1,5 mm. Napětí -5 V pro nedostatek špiček na konektoru (je jich jen 41) není na sběrnici vyvedeno. Je-li na některé desce třeba napětí -5 V, použije se Zenerova dioda, a při větším odběru proudu pak ještě operační zesilovač a výkonový tranzistor (obr. 2), přičemž jako referenční napětí se



Obr. 2. Výkonový zdroj napětí -5 V

používá +5 V. Ostatní špičky na sběrnici propojíme pocínovaným měděným vodičem menšího průřezu (asi 0,5 mm). Pro zadní sběrnici používáme 12kólkový konektor TESLA (WK 462 a WK 465), kterého současně využíváme pro vedení signálů z mikropočítače k periferním zařízením. Jako hlavní konektor je použit původní konektor z kalkulačky ELKA.

Vana Intelky má místo pro 25 desek. Je na vlastním uvážení, umístíte-li zdroj pro napájení mikropočítače mimo vanu, nebo oželíme-li několik posledních de-

sek, vyjmeme konektory a do volného místa zdroj vestavíme. Celé zařízení pak bude tvořit kompaktní celek a vanu Intelky můžeme popřípadě vrátit do kóstry ELKY. Abychom mohli přivést kabely do periferních zařízení, odřízneme zadní stěnu. Původní klávesnici vyjmeme a nahradíme alfanumerickou klávesnicí. Tato klávesnice může být tlačítková nebo i membránová (např. viz JPR-1). Celek pak působí velmi pěkným dojmem a dostane profesionální vzhled. Na vrchní plochu Intelky lze pak umístit televizi pro alfanumerický displej.

Mikroprocesorová deska INTELKA 001

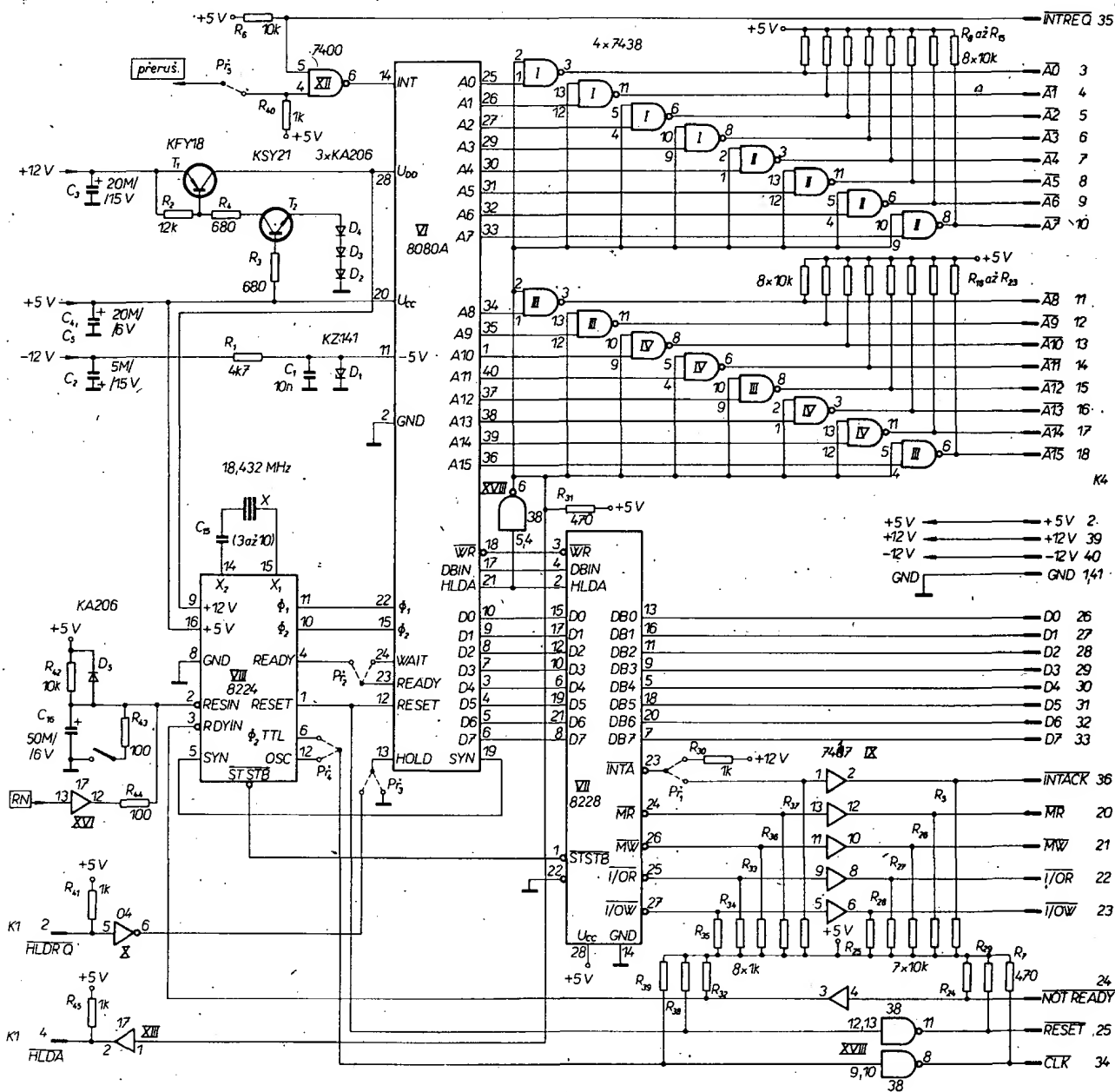
Zapojení na této desce je sestaveno ze dvou samostatných částí – mikroprocesorového systému a obvodů výstupu a vstupu (porty). Byl použit mikroprocesor 8080A, neboť se začal vyrábět i u nás. Je v tradičním zapojení (obr. 3) s obvodem generátoru hodin 8224 a obvodem pro dekódování stavového slova a budiče datové sběrnice 8228. 16 adresových linek z μP 8080A je vedeno přes čtyři

obvody 7438 s otevřeným kolektorem na sběrnici. Druhé vstupy hradel jsou spojeny a buzeny hradlem, na jehož vstup je přiveden signál HLDA (potvrzení požadavku na přerušení) z μP 8080A. Toto zapojení zaručí, že při přímém přístupu do paměti budou výstupy obvodů 7438 zavřeny. Data jsou na sběrnici vedena přímo z obvodu 8228, což postačuje pro středně velký systém. Na každé další desce je však lépe datové signály při vícenásobném použití (např. paměti, několik portů apod.) posílat třístavovými budiči 3216. Výstupy řídicích signálů z obvodu 8228 jsou na sběrnici vedeny přes zesilovače 7417, které je zesilují a chrání před zkratem.

Funkce přepínačů:

Přepínač Pf_1 – při připojení výstupu INTA obvodu 8228 na napětí +12 V přes odpor 1 k Ω obvod 8228 zajistí, že při přerušení je vyslána na datovou sběrnici mikroprocesoru instrukce RST 7. V druhé poloze přepínače je výstup INTACK vyveden na sběrnici a pak periferní zařízení při přerušení práce mikroprocesoru musí samo dodat na datovou sběrnici instrukci RST 0 až RST 7.

Přepínač Pf_2 – pokud v mikropočítači použijeme pomalejší paměti (delší doba přístupu do paměti než 450 ns), je vhodné cyklus paměti (čtení nebo zápis) zpozdit. Mikroprocesor na začátku svého cyklu v době T_1 (náběhu hodin Φ_1) vysílá na adresovou sběrnici adresu paměti, na které je uložena další instrukce. V době T_2 náběhu druhého impulsu hodin Φ_1 mikroprocesor testuje svůj vstup READY. Je-li na něm úroveň H, potom mikroprocesor v době T_3 impulsu hodin Φ_1 čte data z paměti. Je-li na vstupu READY úroveň L, potom mikroprocesor vstupuje do stavu WAIT (čekaj). Tento stav oznámí na svém výstupu WAIT úroveň H. Spojíme-li proto vstup READY a výstup WAIT, bude cyklus paměti zpozděn o jeden hodinový impuls. Na začátku cyklu je na výstupu WAIT úroveň L, proto i na vstupu READY je L. To má za následek vyvolání stavu WAIT, na výstupu WAIT mikroprocesoru je již H. Další hodinový cyklus ukončí stav WAIT (na vstupu READY je H) a mikroprocesor pokračuje v další práci. Při stavu WAIT nemůže nastat přerušení ani DMA. Proto jej nepoužíváme u pomalých periférií (např. u klávesnice apod.).



Obr. 3. Zapojení mikroprocesoru 8080A

Přepínač Pf_3 slouží jen k tomu, aby se ze zadní sběrnice k μP 8080 dostal signál \overline{HLDRO} . V opačném případě je na vstupu \overline{HOLD} mikroprocesoru stále L.

Přepínačem Pf_4 se v jedné jeho poloze dostává signál z oscilátoru na výstup sběrnice CLK (hodiny). Je to tedy signál o kmitočtu krystalu. V jeho druhé poloze se na sběrnici dostává hodinový signál Φ_2 mikroprocesoru.

Přepínač Pf_5 v sepnuté poloze umožňuje, aby se signál ze vstupního obvodu 3212 (portu) dostal na vstup mikroprocesoru INT (je sečten se signálem \overline{INTREQ} ze sběrnice). V opačném případě je na vstup INT mikroprocesoru přiveden jen signál ze sběrnice.

Kmitočet hodin (a tím i rychlost práce mikroprocesoru) lze řídit vnějším připoje-

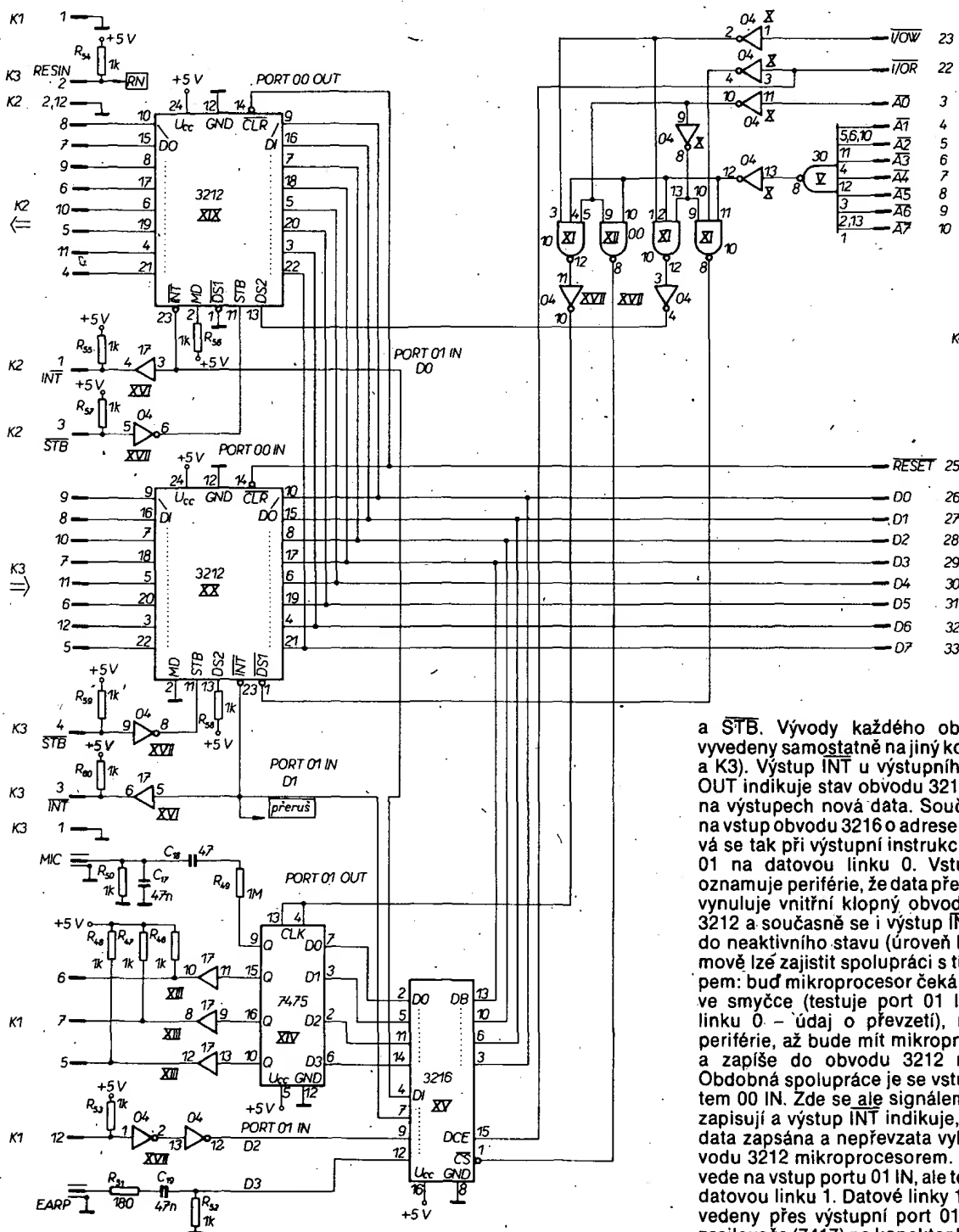
ním krystalu k obvodu 8224. Maximální rezonanční kmitočet krystalu smí být okolo 18 MHz, ale používáme raději nižší. Na přesném kmitočtu krystalu nezáleží, pokud signál oscilátoru nechceme používat dále (např. jako zdroj kmitočtu pro reálný čas). Kmitočet oscilátoru můžeme v malých mezích měnit sériovým kondenzátorem o kapacitě 3 až 10 pF (při kmitočtu krystalu vyšším než 10 MHz).

Nulování mikroprocesoru a mikropočítače je zajištěno pomocí vstupu RESIN obvodu 8224. Na tento vstup je přiveden i signál ze zadního konektoru vstupního portu. Je proto možné nulovat mikropočítač přímo z klávesnice. Dále je na vstup RESIN připojen obvod RC, zajišťující vynulování Intelky po zapnutí napájení, a mikropřepínač, umístěný na desce s plošnými spoji. Tento mikropřepínač dovoluje

nulovat Intelku při ožiovování, kdy není ještě připojena klávesnice.

Napětí -5 V pro μP 8080A se získává z napětí -12 V Zenerovou diodou (malý odběr proudu). Napětí +12 V se na mikroprocesor vede přes tranzistor KFY18, který je buzen z tranzistoru KSY81. Celé toto zapojení zaručí, že při vypnutí napětí +5 V (např. při zkratu) nebude na mikroprocesor a obvod 8224 přivedeno napětí +12 V. Kdyby napájení +12 V zůstalo, obvody by se zničily!

Druhá část desky (obr. 4) obsahuje dekodér adres, vstupní a výstupní osmibitový port a další jednotlivé porty. Výstupní port 00 OUT je tvořen obvodem 3212. Z dekodéru adres je patrné, že je vybrán při adrese 00. Na této adrese je také umístěn vstupní port 00 IN. Porty mají kromě datových i řídicí signály a to INT

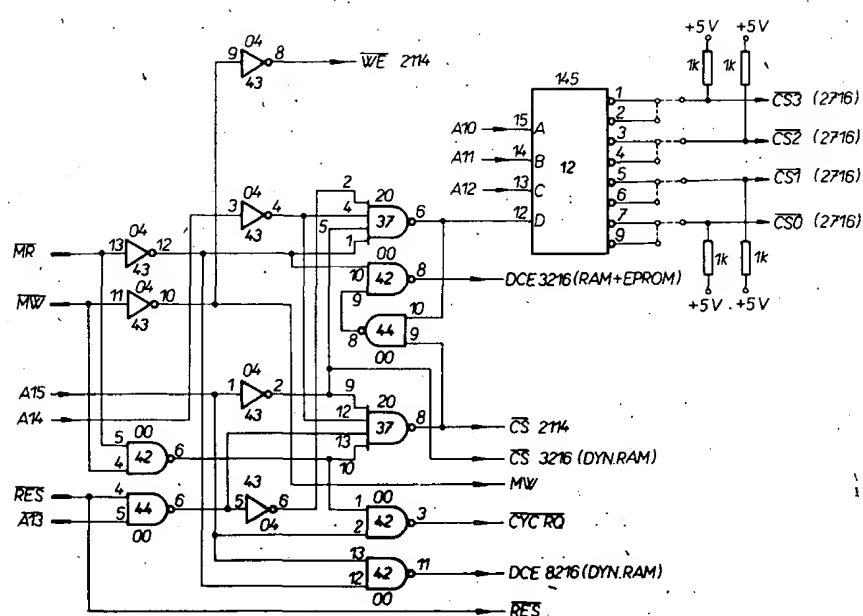


a STB. Vývody každého obvodu jsou vyvedeny samostatně na jiný konektor (K2 a K3). Výstup INT u výstupního portu 00 OUT indikuje stav obvodu 3212, kdy jsou na výstupech nová data. Současně vede na vstup obvodu 3216 o adrese 01 a dostává se tak při výstupní instrukci s adresou 01 na datovou linku 0. Vstupem STB oznamuje periférie, že data převzala, a tím vynuluje vnitřní klopný obvod v obvodu 3212 a současně se i výstup INT dostane do neaktivního stavu (úroveň H). Programově lze zajistit spolupráci s tímto výstupem: buď mikroprocesor čeká na periférii ve smyčce (testuje port 01 IN datovou linku 0 - údaj o převzetí), nebo čeká periférie, až bude mít mikroprocesor čas a zapíše do obvodu 3212 nová data. Obdobná spolupráce je se vstupním portem 00 IN. Zde se ale signálem STB data zapisují a výstup INT indikuje, že jsou již data zapsána a nepřevzata vybráním obvodu 3212 mikroprocesorem. Výstup INT vede na vstup portu 01 IN, ale tentokrát na datovou linku 1. Datové linky 1, 2 a 3 jsou vedeny přes výstupní port 01 OUT přes zesilovače (7417) na konektor K1. Vstupní port 01 IN zahrnuje ještě jeden vstup na datové lince 2. Přes výstupní port 01 OUT (datovou linku 0) a vstupní port 01 (datovou linku 3) je realizován vstup a výstup

mikropočítače na vnější magnetofon, sloužící pro záznam a čtení dat nebo programů. Zapojení s odpory a kondenzátory je převzato z mikropočítače ZX-80.

Paměťová deska INTELKA 002: Tato deska v plném osazení součástkami obsahuje 32K byte dynamické paměti RAM s obvody 4116, 8K byte stálé paměti EPROM s obvody 2716 (nebo 4K byte s obvody 2708) a 1K byte statické paměti RAM s obvody 2114. Záleží na konstruktérovi, čím desku osadí. Obvodově je možno paměťovou desku rozdělit do několika jednodušších celků:

1. **Dekodér adres** (obr. 5) obsahuje několik hradel, invertorů a obvod 74145. Generuje signály pro vybírání jednotlivých druhů pamětí a řídicí signály pro ovládání třístavových zesilovačů dat (obvody 3216). Paměť EPROM a 1K byte RAM statické paměti mají společné zesilovače dat. Vstupy CS těchto zesilovačů (obvody 3216) jsou tedy uzemněny, zesilovače se proto nedostanou do třetího stavu. Pomocí signálu DCE jsou jejich směry přenosu obráceny směrem k pamětem. Pouze při vybrání paměti při adresách od 0000_H do 3FFF_H a současně při čtení z paměti se tyto zesilovače „obracejí“ a předávají data na datovou sběrnici. Aby se do paměti nezapsala data, která jsou na sběrnici, jsou paměť EPROM a 1K byte RAM vybírány pomocí signálu CS. Paměť EPROM je umístěna, jak je patrné z dekodéru adres, v oblasti adres 0000_H až 1FFF_H. Můžeme zde použít paměti 2716 (2K byte EPROM), nebo paměti typu 2708 (1K byte EPROM). Toto uspořádání umožní drátové propojky, které vedou z obvodů dekodéru 74145 (BCD na 1 z 10), jehož osm prvních výstupů je možno libovolně propojit se vstupy CS pamětí EPROM. Při úrovni H na vstupu D u obvodu 74145 nebude aktivní ani jeden z osmi prvních výstupů vedoucích k paměti EPROM, a proto také nebude tato paměť vybrána. Statická paměť 1K byte RAM je umístěna na adrese 2000_H až 3FFF_H. To znamená, že je umístěna osmkrát za sebou a na to musíme dát pozor při zápisu (a čtení) dat



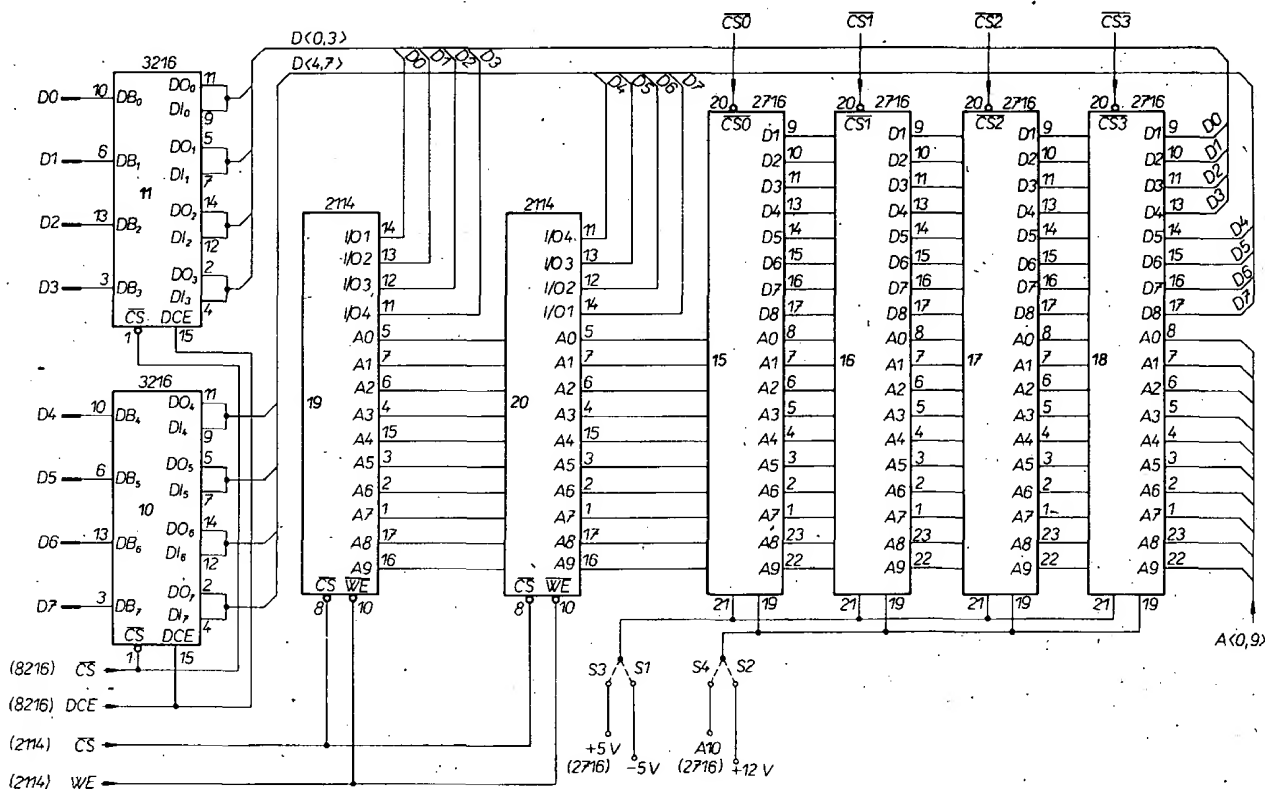
Obr. 5. Dekodér adres

na tyto adresy. Druhých 16K byte paměti (adresy 4000_H až 7FFF_H) je volných. Na toto paměťové místo lze umístit paměť na jiné desce, nebo ho použít na paměť pro televizní displej. Vyšších 32K byte (adresa 8000_H až FFFF_H) je plně obsazeno dynamickou pamětí RAM s obvody 4116. Tato paměť má i své zesilovače dat. Zesilovače jsou stále ve třetímu stavu, teprve při adrese od 8000_H jsou vybrány (na vstupu CS je L). Výstup dekodéru adres CYC RQ oznamuje, zda je dynamická paměť aktivována (čtením nebo zápisem).

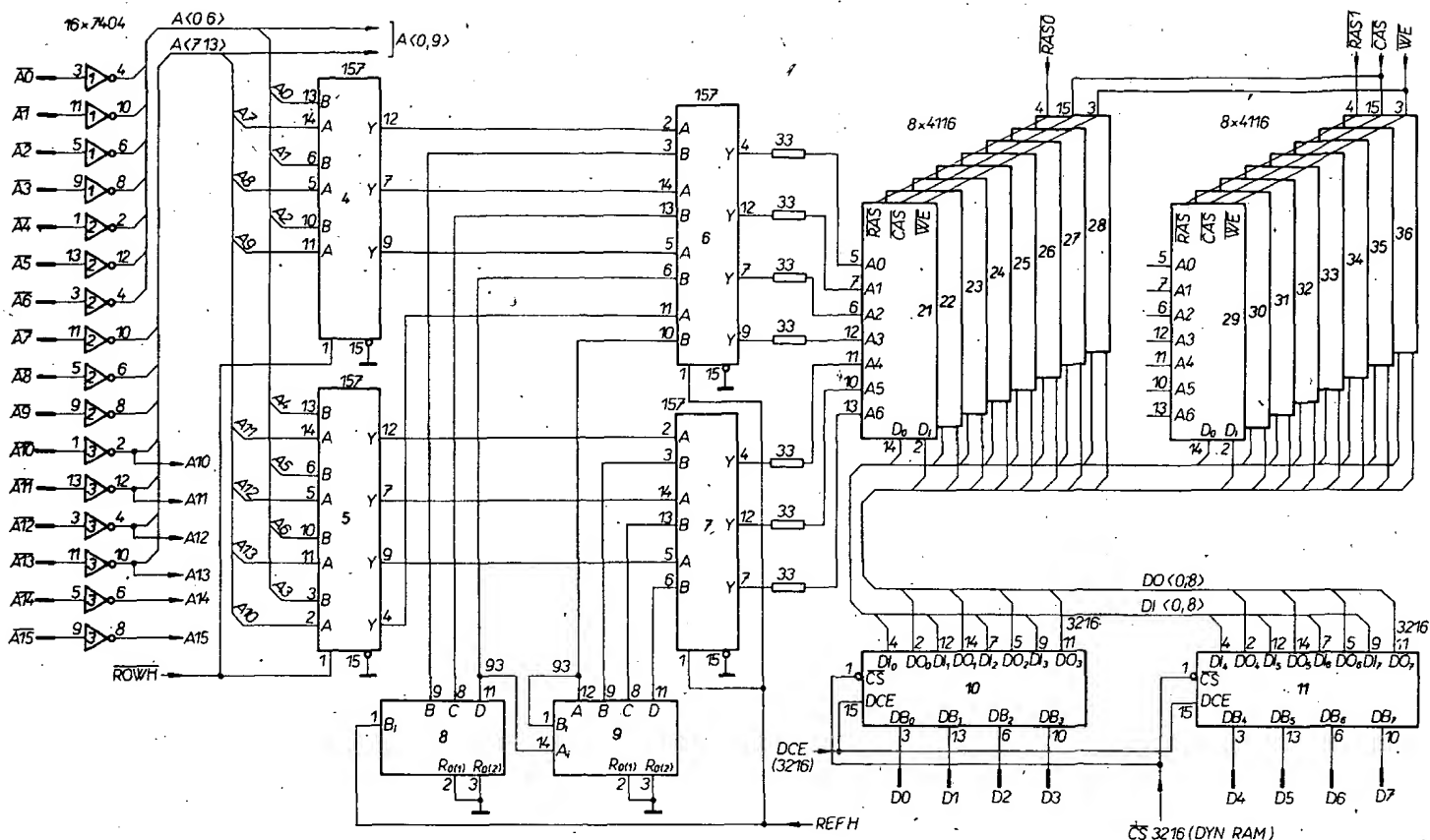
2. **Statická paměť RAM a pevná paměť EPROM** (obr. 6). Jsou použity dva obvody 2114 (1K x 4 bity RAM) a jako pevná paměť EPROM 1 až 4 obvody 2708 nebo 2716. Řídicí a vybavovací signály jsou vedeny z dekodéru adres. Stejně výstupy dat u pamětí EPROM jsou spojeny a přive-

deny na zesilovače 3216 a současně jsou spojeny s vývody statické paměti RAM. Adresy jsou k pamětem přivedeny přes invertory na multiplexery adres (obvody 74157) a jsou přepínány signálem ROW H. Tímto způsobem přepínáme adresu řádku a sloupce u dynamické paměti. Od těchto multiplexerů je vedena adresa na další multiplexery, kde se přepíná s adresou občerstvovací (cyklus REFRES H). Občerstvovací adresu vytvářejí čítače, sestavené z obvodů 7493. Při každém dalším občerstvení paměti je adresa o jednu vyšší, čímž se zajistí občerstvení celé

3. **Obvody dynamické paměti RAM** (obr. 7). Adresy ze sběrnice jsou vedeny přes invertory na multiplexery adres (obvody 74157) a jsou přepínány signálem ROW H. Tímto způsobem přepínáme adresu řádku a sloupce u dynamické paměti. Od těchto multiplexerů je vedena adresa na další multiplexery, kde se přepíná s adresou občerstvovací (cyklus REFRES H). Občerstvovací adresu vytvářejí čítače, sestavené z obvodů 7493. Při každém dalším občerstvení paměti je adresa o jednu vyšší, čímž se zajistí občerstvení celé



Obr. 6. Statická paměť RAM a paměť EPROM

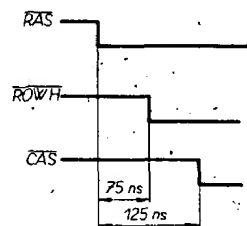


Obr. 7. Obvody dynamické paměti RAM

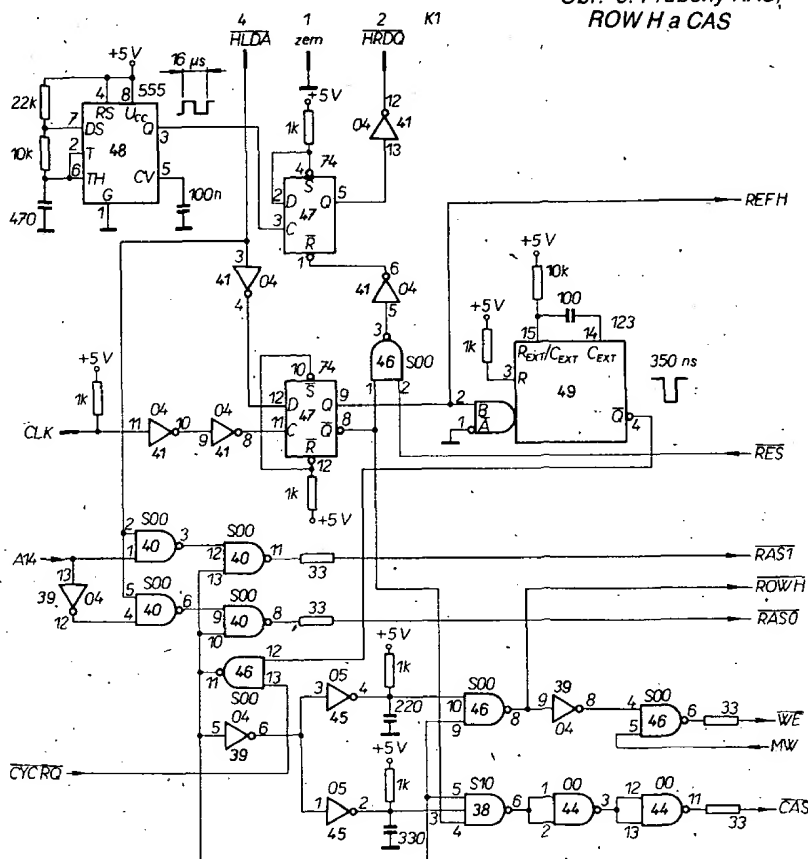
paměti. Multiplexer občerstvovací adresy se přepíná signálem REF H. Výstupy těchto multiplexerů vedou přes odpor 33 Ω na adresové vstupy dynamických paměti 4116. Datové vstupy a výstupy jednotlivých paměti jsou přivedeny přes zesilovače dat 3216 na datovou sběrnici.

4. Vytváření signálů pro občerstvení dynamických paměti RAM je na obr. 8. Obvod 555 generuje časové impulsy o délce 16 μ s, které řídí požadavek na přímý přístup do paměti (nahodí klopný obvod typu D). Po přijmu potvrzení požadavku na DMA (signál HLDA je L) jsou zablokována hradla NAND 40, vstupy 1 a 4. Na výstupech těchto hradel je pak úroveň H a otvírají se hradla 40, vstupy 10 a 13, která jsou spojena s výstupem 11 hradla 46. Toto hradlo počítá impuls monostabilního obvodu 49 (74123) se signálem CYC RQ z dekodéru adres. Z toho je patrné, že při občerstvování bude aktivní horní i dolní řada paměti (RAS 1 i RAS 0 budou totožné). Současně signál HLDA vede přes invertor na vstup D klopného obvodu 47 a zapishe se zde při příchodu hodinového impulsu Φ_2 . Výstup Q obvodu 47 vede na vstup monostabilního obvodu 49, který vyrábí invertovaný impuls o šířce 350 ns. Výstup Q klopného obvodu 47 se počítá se signálem RESET pomocí hradla NAND 46 a nuluje klopný obvod 47 a tím i požadavek na přímý přístup do paměti. Průběhy RAS, ROW H a CAS jsou na obr. 9. Zpoždění se dosáhne zpožďovacími členy s invertory 7405. Při běžném čtení z paměti bude mít signál HLDA úroveň H a tak budou aktivována hradla 40, vstup 1 a 4. V tomto případě bude po příchodu signálu CYC RQ aktivní pouze jedna ze dvou řad paměti (RAS 1 nebo RAS 0) díky invertoru v signálu A14. Při občerstvování paměti signál CAS nebude aktivní díky vstupu Q klopného obvodu

47, na němž je úroveň L. Při zápisu do paměti bude aktivní signál WE, neboť při signálu MW bude mít úroveň H. Při občerstvování paměti je signál MW úrovně L a tak signál WE není aktivní, což znamená čtení z paměti.



Obr. 9. Průběhy RAS, ROW H a CAS



Obr. 8. Vznik signálů pro občerstvování dynamických paměti RAM

5. Zdroj napětí -5 V (obr. 2). Jelikož napětí -5 V potřebují paměti 4116, nemůžeme toto napětí získat vzhledem k velkému odběru proudu Zenerovou diodou. Napětí -5 V se získává z -12 V tranzistorem KD338, který je buzen přes tranzistor TR15 operačním zesilovačem 741. Potřebné referenční napětí se odebrá ze stabilizovaného napětí +5 V.

Osazení desek

Každá deska s plošnými spoji má rozměry 170 x 270 mm. Rozmístění součástek na deskách je na obr. 10 a obr. 11. Obrázek plošných spojů jsou na obr. 12, 13, 14 a 15. Při vrtání děr pro konektor, hlavní sběrnice se musíme přesvědčit, který ze dvou druhů konektorů máme k dispozici. Jeden má vzdálenosti mezi dvěma sousedními špičkami 2,25 mm, druhý 2,3 mm. Všechny ostatní díry vrtáme vrtákem o Ø 0,8 mm, jen pro zadní konektory vrtákem Ø 1,2 mm a pro elektrolytické kondenzátory o Ø 0,9 mm. Nejprve si označíme, kam přijdou jednotlivé součástky a propájíme všechny propojky z jedné strany desky na druhou. Pájíme důkladně, zvláště propojky, které jsou umístěny pod integrovanými obvody. Použijeme tenký pocínovaný drát, jehož konce z každé strany desky ohneme a pak pájeme. Pak připevníme všechny konektory. V této fázi práce se vyplácí vztít ohmmetr a desku přeměřit z hlediska zkratů a přerušení spojů. Někomu se to může zdát zbytečné, ale ušetří to později mnoho práce s hledáním závad.

Dále osadíme desku integrovanými obvody, objímkami pro ně a ostatními součástkami. Pro filtrování napájecího napětí použijeme keramické kondenzátory 10 až 47 nF. Při použití paměti EPROM (obvodů 2708) musíme osadit kondenzátory C, 47 nF. Použijeme-li obvody 2716, kondenzátory C, 47 nF neosazujeme (byly by na adresové lince A10). Je vhodné alespoň dražší obvody umístit do objímek (a u paměťové desky obvody 3216). Některé součástky jsou pájeny z obou stran, proto musíme dávat pozor při jejich zapojování. Všechny integrované obvody jsou však pájeny zespodu i za cenu většího množství propojek. Má to výhodu při případném odstraňování vadného zapájeného obvodu z desky.

Přepínače na desce realizujeme buď špičkami z konektoru FRB nebo přímo kousky drátu. Proto je lépe předem uvážit, jak je chceme nastavit. Při spojení obou desek je nutné, aby přepínač P₁ umožňoval signál HLDRQ a přepínačem P₂ se dostával signál hodin Φ₂ na sběrnici. Použijeme-li jako paměť EPROM obvody 2708, spojíme spojky S1 a S2. Spojky S3 a S4 zůstanou rozpojeny! U obvodů paměti EPROM 2716 je tomu naopak.

Neseženeme-li obvody 74145, můžeme je nahradit obvodem 7442. Jelikož každý jeho výstup dekoduje 1K-byte paměti, nemusíme při použití paměti 2708 dělat žádné úpravy. Při použití paměti 2716 (2K byte) nemůžeme však jednoduše spojit dva výstupy obvodu 7442, jak tomu bylo u obvodu 74145 (otevřený kolektor). Pro naše účely však stačí sečíst signály diodami (např. KA206), jejichž anody budou spojeny se vstupem paměti CS a katody přivedeny k výstupům obvodu 7442. (Pro každou paměť 2716 je tedy zapotřebí dvou diod.) Protože používáme sběrnici s obvody s otevřeným kolektorem, je nutné použít zakončovací odpory (obr. 16). Tyto odpory nejlépe umístíme přímo na sběrnici a druhým koncem na pomocnou lištu napájení +5 V. Když si prohlédneme zapojení mikroprocesoru

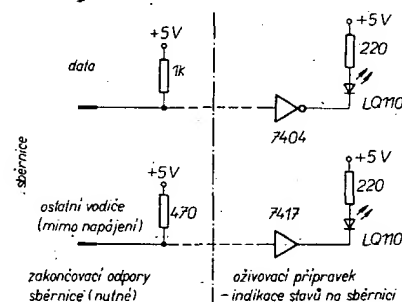
(obr. 3), vidíme, že také tam jsou odpory a to paralelně k zakončovacím odporům. Jsou to odpory pro měření a ožívání, při němž je deska mikroprocesoru umístěna mimo sběrnici. Tyto odpory nemusíme proto osadit, případně můžeme zmenšit jejich hodnoty a tak umístit zakončovací odpory na mikroprocesorové desce. Budou tam však chybět odpory 1 kΩ na datových linkách.

Oživení desek

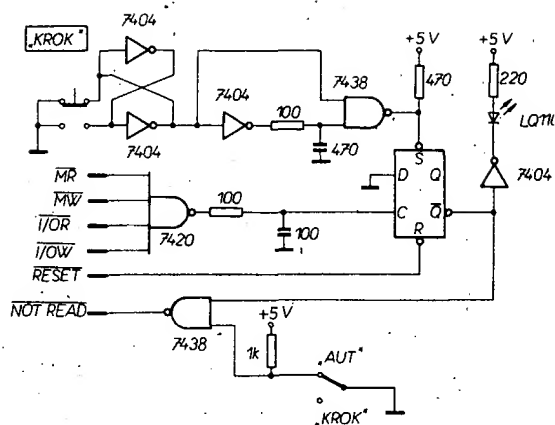
Nejprve zkontrolujeme napětí zdroje -5 V a u mikroprocesorové desky funkci ochranného spínače +12 V. Před zasunutím integrovaných obvodů do objímek je dobré se přesvědčit měřením, jsou-li napájecí napětí skutečně pouze na odpovídajících špičkách objímek (především napětí +12 V a -5 V). Napájecí napětí pro hlavní obvody jsou v tab. 1. Všechny tyto práce předpokládají hotový funkční zdroj (popř. laboratorní zdroj). Změříme odběr proudu a zjistíme, nejsou-li na napájecích napětech zkrat. Zasuneme obvody do objímek (kromě integrovaných obvodů 8228 a μP 8080). S úspěchem můžeme použít při ožívání přípravku TST-03 z JPR-1; potom místo mikroprocesoru a obvodu 8228 zasuneme zástrčku přípravku. Logickou sondou prozkoušíme ve statickém režimu vstupní a výstupní obvody, paměť EPROM a statickou paměť RAM – zkusíme tedy obě desky společně. Pak můžeme zasunout do objímek mikroprocesor 8080 a obvod 8228. Do objímky paměti ROM zasuneme zástrčku přípravku TST-02, do něhož zaznamenejme jednoduchý program (např. skok na nulovou adresu apod.). Je vhodné si k tomu účelu postavit přípravku na krokování mikroprocesoru (obr. 17), případně jej doplnit o signalizaci stavů na sběrnici (obr. 16).

Nakonec se pustíme do ožívání dynamické paměti RAM. Podle osciloskopu nastavíme požadované průběhy jednotlivých signálů (obr. 9). Po nastavení všech průběhů můžeme zkusit zapsat a číst z dynamické paměti. My jsme k tomu použili simulátor paměti EPROM, který bude také uveřejněn v AR. Tento simulátor obsahuje paměť RAM, do níž lze zapsat jednoduchou klávesnicí program. Po napsání programu se tato paměť RAM chová jako paměť EPROM s pevným programem (simulátor má vlastní oddělené napájení). Postavili jsme si také přípravku na obr. 18, který má jednoduchý osmibitový výstup a další tři jednotlivé jednobitové výstupy a jeden jednobitový vstup. K práci s přípravkem bude sloužit několik jednoduchých programů, např. program pro rotaci jednoho bitu pomocí mikrosplínače (tab. 2). Změníme-li adresy

Obr. 10 až 15 jsou na str. 16 až 21



Obr. 16: Zakončovaci odpory



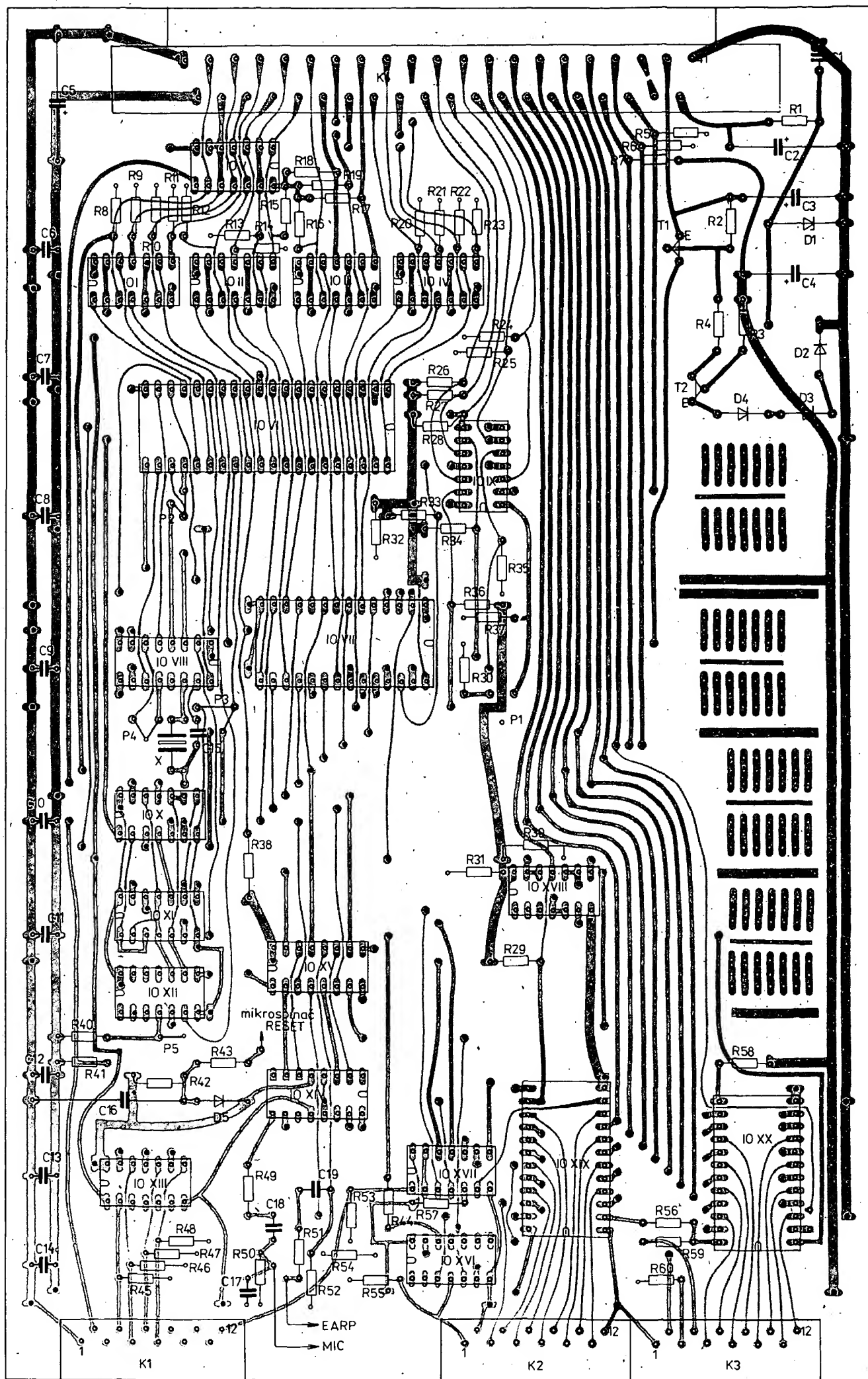
Obr. 17: Přípravek ke krokování mikroprocesoru

Tab. 1. Vývody IO, na něž se přivádějí napájecí napětí

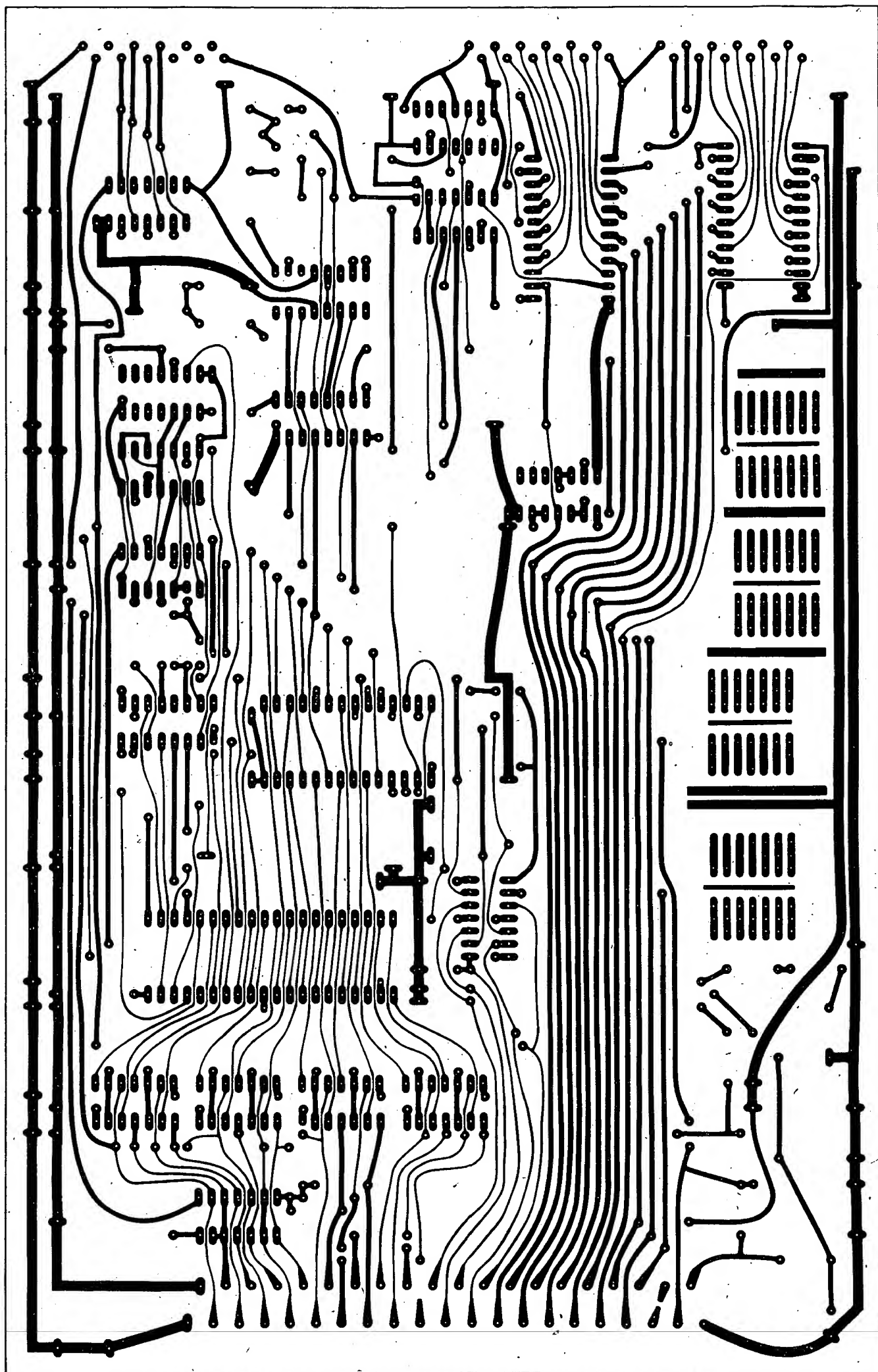
| Obvod | +5 V | +12 V | -12 V | -5 V | 0 V |
|-------|--------|-------|-------|------|--------|
| 8080A | 20 | 28 | | 11 | 2 |
| 8228 | 28 | | | | 14 |
| 8224 | 16 | 9 | | | 8 |
| 3216 | 16 | | | | 8 |
| 3212 | 24 | | | | 12 |
| 2716 | 24, 21 | | | | 12, 18 |
| 2708 | 24 | 19 | | 21 | 12, 18 |
| 2114 | 18 | | | | 9 |
| 4116 | 9 | 8 | | 1 | 16 |

Tab. 2. Rotace bitu pomocí mikrosplínače

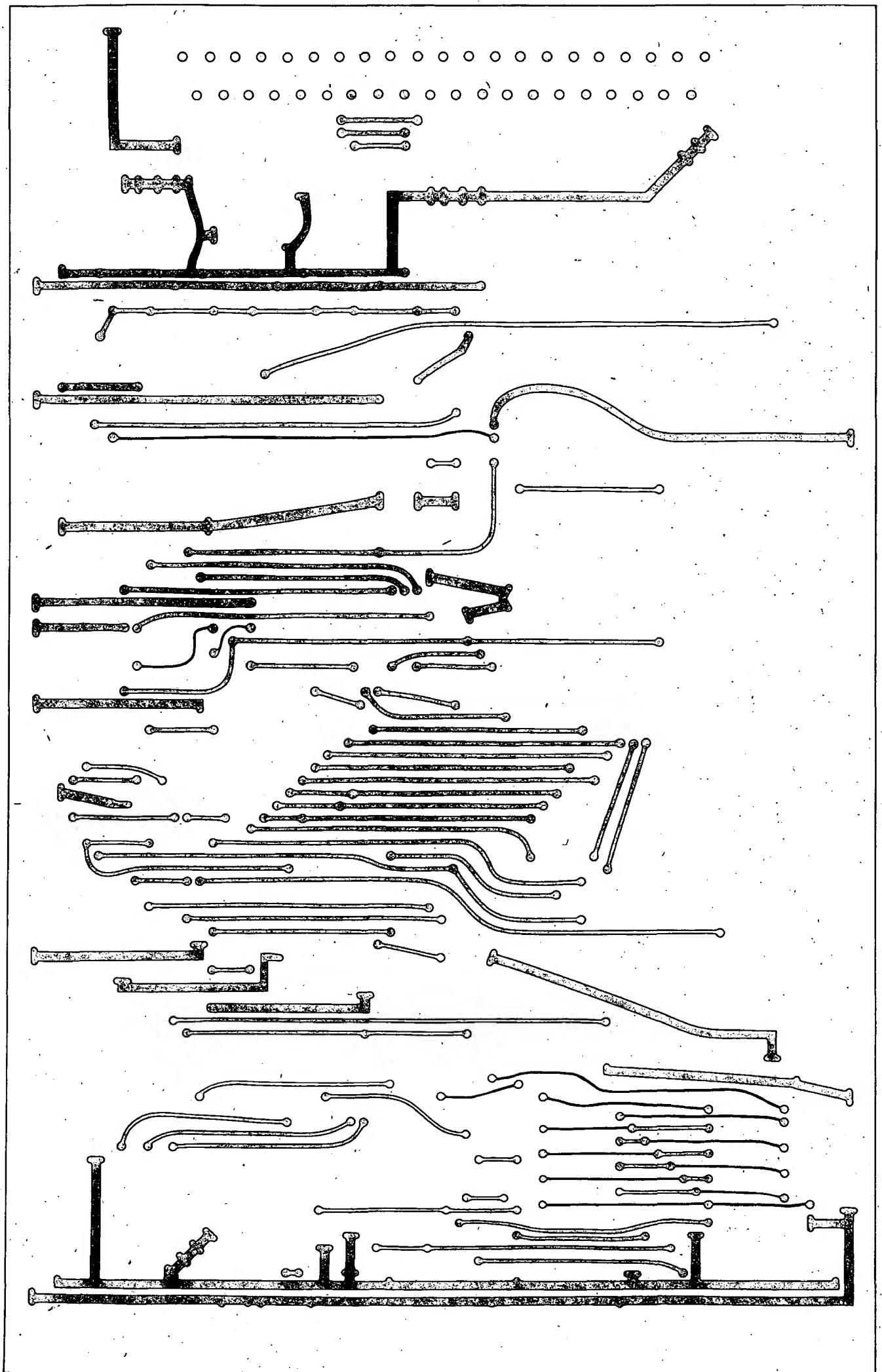
| Adresa | Data | Mnem. kód | Poznámky |
|--------|------|-----------|--|
| 00 | 1E | MVI E | zapsání dat 01 do registru E |
| 01 | 01 | 01 | |
| 02 | 7B | MOV A, E | přemístění registru E do střídače |
| 03 | D3 | OUT | obsah střídače na výstupní port o adrese 00 |
| 04 | 00 | 00 | rotace obsahu střídače |
| 05 | 07 | RLC | |
| 06 | 5F | MOV E, A | přemístění střídače do registru E |
| 07 | DB | IN | vstup z portu 01 do střídače |
| 08 | 01 | 01 | násobení AND obsahu střídače s daty 01 – maskování – bit 0 |
| 09 | E6 | ANI | |
| 0A | 01 | 01 | skok na adresu 0007 při nenulovém obsahu střídače |
| 0B | C2 | JNZ | |
| 0C | 07 | 07 | skok na adresu 0002 |
| 0D | 00 | 00 | |
| 0E | C3 | JMP | skok na adresu 0002 |
| 0F | 02 | 02 | |
| 10 | 00 | 00 | |

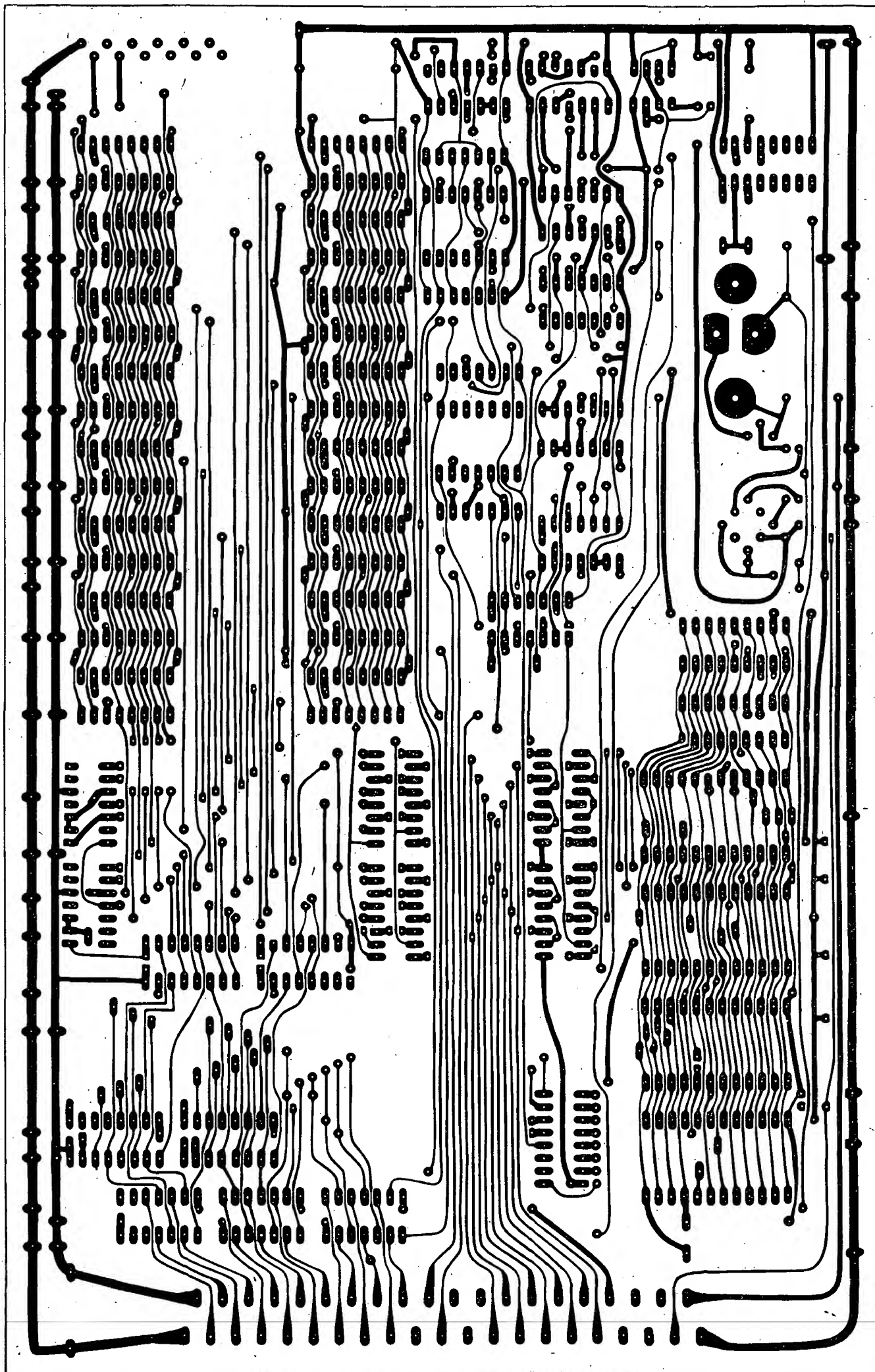


Obr. 10. Osazení mikroprocesorové desky součástkami

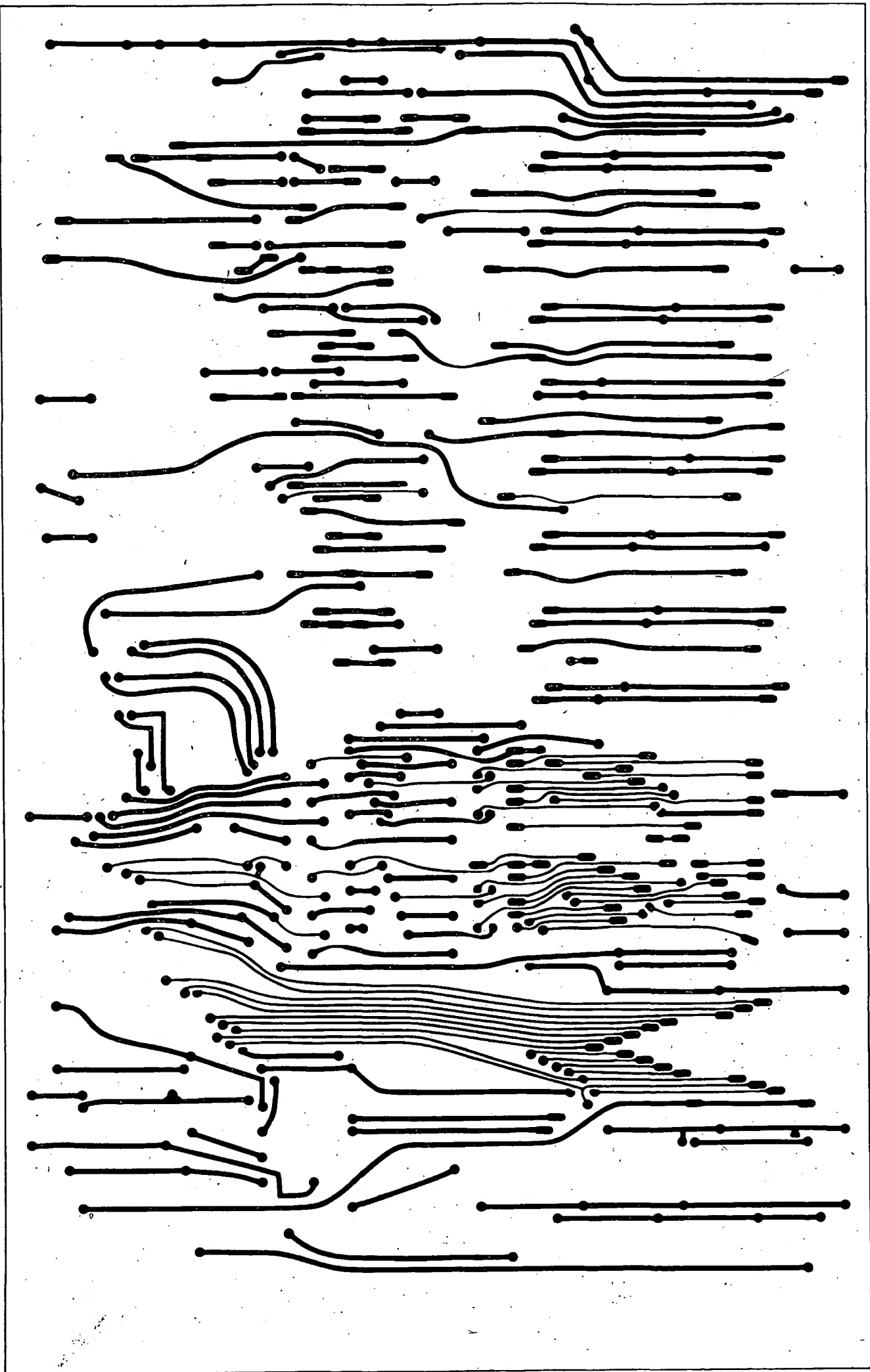


Obr. 12, 13. Deska s plošnými spoji R 201 (mikroprocesorová deska Intelka 001)

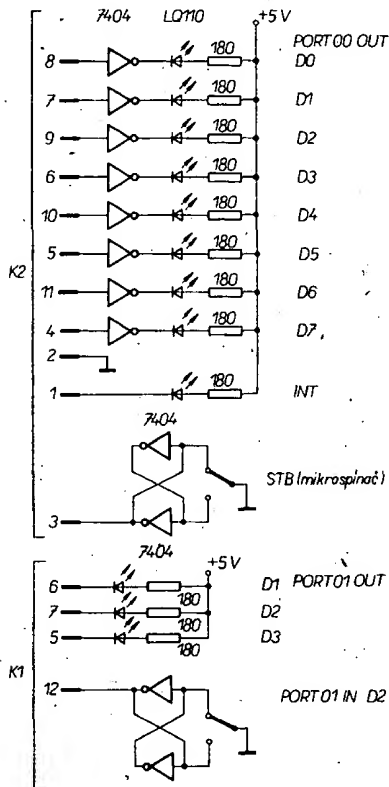




Obr. 14, 15. Deska s plošnými spoji R 202 (deska paměti 002)



tohoto programu tak, aby mohl být uložen v dynamické paměti a zapíšeme-li ho tam, tak poslouží jako dobrá kontrola funkce této paměti. Byla-li by paměť nesprávně občerstvována, program asi za 6 s po stisknutí mikrospínače z paměti zmizí. Do dynamické paměti ho můžeme zapsat např. simulátorem paměti EPROM. Jiný test paměti (nejprve se zapíše do celé paměti 00_h a pak se kontroluje, je-li zápis v pořádku apod.) si napíše jistě každý sám. S velkou výhodou lze k tomu použít přípravek (obr. 18), u něhož ještě



Obr. 18. Přípravek pro ožívání

např. další tři výstupy budou signalizovat, že paměť byla přečtena nesprávně a například, že je na výstupní osmibitový port zapsána nejprve nižší a pak vyšší adresa chybného místa v paměti.

Popsané obvody jsou základem Intelky. Obě desky pracují i bez dalších desek, přitom však další vstupní a výstupní porty si může navrhnout každý sám z obr. 4, je však třeba změnit adresy portů (např. prohodit adresovou linku A0 s adresovou linkou A4 apod.). Na mikroprocesorové desce je také „univerzální místo“, na němž se mohou realizovat případné změny zapojení. Intelku lze doplnit dalšími deskami paměti, např. s obvody 2102 a různými typy převodníků, televizním alfanumerickým displejem apod. Návrh a popis těchto obvodů však již přesahuje rámec tohoto čísla AR řady B.

Na závěr bychom chtěli poděkovat především ing. E. Smutnému a ing. V. Krausovi a samozřejmě i všem ostatním, kteří neváhali obětovat svůj volný čas a byli nápomocni při stavbě mikropočítače INTELKA.

Seznam součástek pro mikroprocesorovou desku 001

| Polovodičové součástky | | | |
|------------------------|-------|----------------------------------|-------|
| T ₁ | KFY18 | D ₁ | KZ141 |
| T ₂ | KSY21 | D ₂ až D ₅ | KA206 |

Integrované obvody

| | |
|---------------|-------------|
| I, II, III, | |
| IV, XVIII | 7438 |
| V | 7430 |
| VI | 8080A |
| VII | 8228 |
| VIII | 8224 |
| IX, XIII, XVI | 7407 (7417) |
| X, XVII | 7404 |
| XI | 7410 |
| XII | 7400 |
| XIV | 7475 |
| XV | 8216 (3216) |
| XIX, XX | 8212 (3212) |

Odpory

| | | | |
|--|--------|---|-------|
| R ₁ | 4,7 kΩ | R ₂₉ , R ₂₉ , R ₃₈ | |
| R ₂ | 12 kΩ | až R ₄₁ , R ₄₅ až | |
| R ₃ , R ₄ | 680 Ω | R ₄₈ , R ₃₂ | |
| R ₅ , R ₆ , R ₈ | | až R ₃₇ , R ₅₀ , | |
| až R ₂₈ , R ₄₂ | 10 kΩ | R ₅₂ až R ₆₀ | 1 kΩ |
| R ₇ , R ₃₁ , | 470 Ω | R ₄₄ | 100 Ω |
| | | R ₄₉ | 1 MΩ |
| | | R ₅₁ | 180 Ω |

Seznam součástek pro paměťovou desku 002

Polovodičové součástky

| | |
|-------------------|--------------|
| 2716 (2708), 4 ks | 7474 |
| 2114, 2 ks | 7493, 2 ks |
| 4116, 16 ks | 74S00, 2 ks |
| 3216, 4 ks | 74S10 |
| 74123 | 74145 (7442) |
| NE555 | 74157, 4 ks |
| 7400, 2 ks | TR15 |
| 7404, 6 ks | KD338 |
| 7405 | MAA741 |
| 7420 | |

Odpory

| | |
|-------------|-------------|
| 10 kΩ, 4 ks | 220 Ω |
| 4,7 kΩ | 100 Ω |
| 1 kΩ, 11 ks | 33 Ω, 11 ks |

Kondenzátory

| | |
|-----------------|--------------|
| 100 n, 17 ks | 470 pF |
| 50 μF/15 V | 150 n |
| 20 μF/15 V | 47 nF, 22 ks |
| 20 μF/6 V, 2 ks | 22 nF, 9 ks |
| | 100 pF |

III. Mikropočítačový systém JPR-1

Ing. Eduard Smutný

Na následujících stránkách bych chtěl čtenáře AR seznámit s mikropočítačovým systémem JPR-1. Ten, kdo pracuje v oboru minipočítačů nebo na jejich aplikacích v různých odvětvích našeho průmyslu, bude asi zkratku JPR již znát. JPR znamená Jednotka Programového Řízení a řada minipočítačů JPR-12, JPR-8, JPR-12R, JPR-13 (a mikropočítač JPR-80) byla vyvíjena ve spolupráci TESLA Elstroj a TESLA Strašnice a výrobně realizována.

Pro mikropočítač JPR-1 je název *jednotka programového řízení* skutečně výstižný. JPR-1 je skutečně určena spíše pro řízení než k počítání. Řídící jednotka se od počítačů prakticky neliší, pouze jsou v ní kladeny nároky na vyšší spolehlivost, jednoduchost, nízkou cenu a charakter připojených přídavných zařízení. Řídící jednotka musí být řešena stavebnicově, aby bylo možno sestavit jednoduché i složité systémy při dodržení co nejlepších technickoekonomických parametrů. Protože jsem se podílel na vývoji všech jednotek JPR, vím a chtěl bych následujícím popisem ukázat, že vývoj počítačů není práce příliš složitá, zejména nyní, kdy jsou k dispozici mikroprocesory a další složité obvody. Velkou část práce již za nás totiž udělali „součástkáři“, a tak je jen nutné činnosti obvodů porozumět a složit z nich počítač.

Mikropočítač, jak se říká počítači, jenž využívá jako základní jednotky mikroprocesor, je možno navrhovat různě. Nejznámější skupinou jsou tzv. osobní mikropočítače. Ty bývají obvykle řešeny jako konstrukční celek, a to proto, že je známé, co vše má osobní počítač umět. Můžeme pak dát většinu součástek na jednu velkou desku a tím ušetřit počet konektorů, které by musely spojit větší počet malých desek. Pro rozšíření osobního počítače pak slouží buď omezený počet konektorů uvnitř počítače nebo rozšiřovací jednotka (expander), do níž se umísťují desky pro další přídavná zařízení. Výhodou tohoto konstrukčního řešení je nízká cena a nevýhodou nutnost používat celý počítač, i když všech jeho funkcí pro danou aplikaci nevyužijeme – proto se osobní počítače používají spíše na zpracování dat, textů, grafických úloh apod. Pro řízení nejsou příliš vhodné. Označení „osobní“ není však dnes zcela přesné. I když byly tyto počítače určeny původně především pro zájmovou činnost, používají se dnes jako kancelářské počítače, počítače pro přípravu programů pro obráběcí stroje, v konstrukčních kancelářích apod.

Další skupinu tvoří mikropočítačové systémy stavebnicového charakteru, vycházející obvykle z tzv. jednodeskových mikropočítačů. Příkladem může být systém SBC firmy INTEL. Tyto systémy jsou obvykle řešeny tak, že vlastně nahrazují větší a dražší minipočítačové systémy. Mají složité sběrnice, které umožňují spolupráci několika procesorů a rychlé přenosy dat typu DMA. V současné době se ve světě vyrábějí 8, 16 a 32bitové systémy, které jsou velice výkonné, ale také poměrně drahé a náročné na znalosti těch, kteří je programují a aplikují.

Mikropočítačový systém JPR-1 je představitelem třetí skupiny, a to malých stavebnicových systémů s omezenými možnostmi, vyznačujících se jednoduchostí a nízkou cenou. Kdo sleduje vývoj mikropočítačů ve světě, jistě si všiml, že se po první, několik let trvající vlně maximálních požadavků na mikropočítačové systémy, znovu začaly objevovat malé systémy, obsahující jen to nejnútnejší pro danou aplikaci.

Aby byl výčet úplný – poslední a možná největší skupinou jsou mikropočítače „sítě na míru“ a „základní“ v různých hračkách, měřicích přístrojích, sportovním zboží, psacích strojích a dnes vlastně téměř ve všem. Jedním z příkladů je automat na prodej jízdenek na Hlavním nádraží v Praze nebo pokladny v automatu Lucerna. Tyto systémy se používají všude tam, kde je předpoklad velké seriovosti výroby a požaduje se nízká cena a minimální servis. Spolehlivost těchto systémů je dána tím, že jsou v nich pouze nejnútnejší součástky, minimum spojující vše optimálně navazuje na sebe.

Mikropočítač JPR-1 není osobní počítač – na to je příliš „draze“ navržen. V současné době, kdy potřebujeme, aby mikropočítače začaly pomáhat v průmyslu, službách, zdravotnictví a zemědělství, by ani nebylo rozumné dělat osobní počítače, které by si mohl málokdo koupit domů. Mikropočítač JPR-1 je vlastně minimálním systémem s mikroprocesorem 8080A. Má vše, co by počítač měl mít a nemá to, co nezbytně mít nemusí.

Možná že pro ty, kteří si chtějí mikropočítač postavit pouze pro sebe, bude určitým zklamáním, že je nutné mít prokovené díry na deskách s plošnými spoji, drahé konektory a moc a moc součástek, ale nedá se nic dělat. JPR-1 není určen pro amatéry, ti si mohou postavit Intelku, ale pro velkou část techniků v našem průmyslu, kteří chtějí začít s mikroprocesory pracovat, a pro kutily svazarmovských organizací. Aby byla alespoň těmto zájemcům dána možnost si JPR-1 postavit, zvolil jsem konektory FRB (aby nebylo nutno zlatit desky) a co nejjednodušší zapojení desek a zdrojů. Přídavná zařízení jsem zvolil tak, aby byla dostupná a návod na zhotovení klávesnice umožní nezávislost na dodávkách „přídavných zařízení“, s nimiž jsou u nás zatím potíže.

Při dalším vývoji systému JPR-1 se možná ještě něco změní na konstrukci nebo zapojení. Připravil počítač pro sériovou výrobu (a o tom se jedná) je složitější, než se zdá. A tak tedy čtěte a chcete-li, držte palce, ať je JPR-1 brzy na světě a ať si mikro počítač můžete koupit a ne pracně dělat.

SBĚRNICE ARB-1

Mikro počítač JPR-1 byl původně navržen jako jednodeskový, dokonce výstižněji řečeno, jako jednočipový. Jednodeskový počítač má obvykle na desce paměť RAM a EPROM a několik paralelních a sériových portů. Většina jednodeskových mikro počítačů má však vyvedeny všechny signály nutné pro rozšíření systému o další desky. Oproti tomu jednočipový počítač (zanedbáme-li to, že je vyroben v jednom pouzdru) potřebuje ke své činnosti pouze napájecí napětí. Má vlastní paměti a porty a na rozdíl od jednodeskového mikro počítače nelze jednoduše jeho systém paměti a portů rozšiřovat. Původně i JPR-1 měl na konektoru kromě portů pouze napájecí napětí a nebylo možno připojit další desky. Adresové výstupy mikroprocesoru 8080A by ani nebyly schopny budit vstupní přijímače a dekódery sloužící k adresování dalších desek. Po prvních úvahách o způsobu oživování a testování desky JPR-1 jsem vyvedl na konektor adresové, řídicí a datové signály. Pomocí tohoto konektoru je možno desku otestovat, krokovat programy nebo přečíst obsah paměti RAM a EPROM při ladění programů – to už byl vlastně první krok k tomu, aby se zrodila sběrnice, kterou jsem nazval ARB-1. Tato sběrnice byla určena pro toto číslo AR a proto dostala název AR BUS. Co to vlastně sběrnice je, jak pracuje a jaké signály tvoří? Pokusím se to vysvětlit a pak se zase vrátíme ke sběrnici ARB-1.

Sběrnice

Sběrnice neboli BUS je v technice mikro počítačů snad nejpoužívanější slovo. Již samotný čip, mikroprocesor, paměť nebo periferní obvod mají svoje sběrniceové vstupy a výstupy. Při používání tohoto slova je nutné mít na paměti, že každá sběrnice není jen soustavou několika vodičů, označených názvy signálů. Každou sběrnici je nutno především přesně definovat, definice obsahuje úroveň signálů a zátěže přijímačů i vyslačů připojených na sběrnici, logické souvislosti mezi signály, časové parametry signálů, přízpusobovací nebo zakončovací odpory i mechanické parametry. Podle skupin signálů dělíme obvykle sběrnice na adresové, datové, řídicí a napájecí. Podle způsobu přenosu informací po sběrnici dělíme sběrnice na paralelní, sériové, sérioparalelní a multiplexované. Podle začlenění sběrnic do systému je dělíme na systémové, vnitřní, lokální, sběrnice přídatných zařízení atd. Problematika sběrnic není tedy jednoduchá. Velice snadno se sběrnice kreslí v blokovém diagramu, hůře se však stanovuje, pracuje-li správně, nejsou-li překročeny zátěže obvodů budících sběrnici, nedochází-li ke konfliktům při časování signálů a tím vlastně, jaká je spolehlivost přenosu informací po sběrnici.

Sběrnice nejsou vlastně nic nového. I tranzistorové počítače a minipočítače s obvody TTL měly své sběrnice. Jedna definice platila u těchto počítačů několik let. Složitost mechanické konstrukce, technologie zapojování vodičů a pracný návrh obvodů spolupracujících se sběrnici – to byly hlavní důvody, proč se sběrnice

ce nerodily tak rychle jako nyní. Nejrozšířenější minipočítači minulého desetiletí byly typy řady PDP8 americké firmy DEC. Když se dnes podíváme na popis signálů tehdejší typické sběrnice (např. typu PDP8/I), nemůžeme se divit tomu, že si každý raději desku interface koupil než udělal. První vlastností mezi sběrnici byl OMNIBUS, použitý u typu PDP8/E. Tato sběrnice měla 96 signálů, rozvedených ke konektorům matiční desky s plošnými spoji. Pak přišla sběrnice UNIBUS stejné firmy, která byla určena pro 16bitové minipočítače PDP-11. I když měly tyto počítače větší délku slova, stačilo pro tuto sběrnici již jen 56 signálů. Nejrozšířenější sběrnici současných mikro počítačů je MULTIBUS firmy Intel. Tato sběrnice vyšla svou koncepcí právě ze sběrnice UNIBUS. Každá sběrnice mikro počítačových systémů je ovlivněna tím, jaké signály má použitý mikroprocesor. Vznikly tak sběrnice šité na míru pro 8080A, M6800 nebo 6502. Řekneme si tedy nejprve něco o vstupních a výstupních signálech mikroprocesorů, jejichž počet vývodů je obvykle omezen na 40.

Vstupní a výstupní signály mikroprocesorů

První skupinu signálů tvoří napájení. Sem patří vývod „země“ a minimálně jedno napájecí napětí. Mikroprocesor 8080A, který budeme používat v převážné většině, potřebuje tři různé napájecí napětí stejně jako paměti EPROM 2708 nebo DRAM 4116.

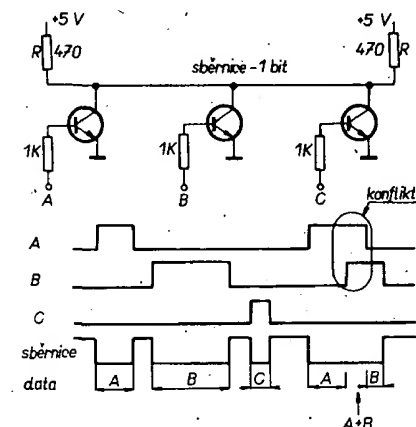
Další skupinu tvoří signály adresy. Mikroprocesory zpracovávají program, který je uložen v paměti. A protože musí nejen něco spočítat, ale je nutné také data do počítače vložit a výsledky „vyndat“ ven, je nutné adresovat i omezený počet vstupních a výstupních přídatných zařízení. Kapacita paměti, to je počet slov, které můžeme maximálně do paměti zapsat nebo z ní přečíst, určuje počet adresových linek, které adresová část sběrnice má. Obvykle má mikroprocesor 16 adresových linek a může tedy adresovat 65 536 paměťových míst (2^{16}). Počet adresovatelných přídatných zařízení bývá menší a využívá se adresy o délce 8 bitů, takže je možno adresovat 256 zařízení (2^8). Posíláme-li „něco“ do výstupního zařízení nebo čteme-li „něco“ ze vstupního zařízení, nepotřebujeme v téměř okamžiku adresovat paměť. Těto skutečnosti využívají mikroprocesory buď přímo tak, že nemají vůbec speciální instrukce pro práci s přídatnými zařízeními a s každým registrem nebo stavovým slovem zařízení se pracuje jako s buňkou paměti, nebo alespoň používají část adresových linek k adresaci přídatných zařízení. Dopusť se nedá říci, co je výhodnější. Dekódování plně 16bitové adresy pro přídatná zařízení je náročnější na počet obvodů a se zvětšováním kapacity paměti na jednom čipu (dnes 64K DRAM) pak přistupuje ještě nutnost zablokovat funkci paměti v těch místech, kde jsou adresy přídatných zařízení.

Instrukce pro vstup a výstup jsou obvykle kratší (2 byte) než instrukce pro čtení a zápis do paměťové buňky (3 byte), ale instrukce pracující s pamětí jsou zase chytřejší. Návrhář systému si může u většiny mikroprocesorů vybrat, zda bude pro přídatná zařízení dekodovat adresu paměti (16 bitů), nebo adresu pro vstup a výstup (8 bitů). Obvykle se adresace přídatných zařízení jako paměť (memory mapped I/O) používá u systému, u něhož se nepočítá s plnou pamětí. Ve skupině řídicích signálů je u většiny mikroproce-

sorů signál, který říká, že na adresové sběrnici je adresa paměti nebo přídatného zařízení. U mapovaných adres se tento signál vytváří dekodováním nejvyšších adresových linek.

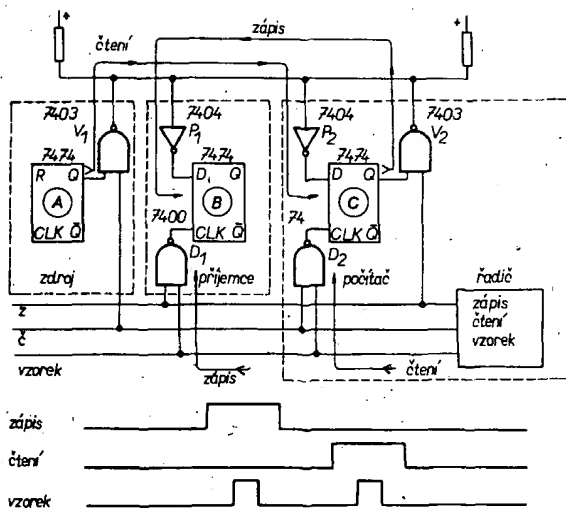
Nejdůležitější skupinou signálů jsou data, neboť jejich zpracování je základem počítačů. Z pouzdra mikroprocesoru vycházejí jako obousměrné linky, aby se ušetřil počet potřebných vývodů. Systémové sběrnice mikro počítačů pak mohou mít zvlášť datové linky vstupu a zvlášť datové linky výstupu. U mikroprocesorů, s nimiž budeme pracovat, se používá 8 datových linek. Říkáme proto, že mají délku slova 8 bitů.

Obousměrná sběrnice dat byla již u minipočítačů zcela běžná. Data se přenášela po linkách, k nimž se zdroje dat (registry počítače, paměti, registry přídatných zařízení) připojovaly přes hradla s otevřenými kolektory (7438). Tím, že v daném okamžiku byl ke sběrnici připojen pouze jeden zdroj signálu, byla zajištěna možnost obousměrného přenosu dat po sběrnici. Základem těchto sběrnic byla vlastně funkce NEBO (OR), vzniklá připojením několika kolektorů budících tranzistorů k jednomu vodiči sběrnice. Na obr. 1 je



Obr. 1. Základní princip sběrnice

základní princip sběrnice. Je to vlastně multiplexovaná sběrnice, neboť jsou na ní v různých okamžicích platná data z různých zdrojů určená pro různé příjemce. Nahradiť-li vyslače sběrnici hradla s otevřenými kolektory a dokreslíme-li zdroje a příjemce dat, dostaneme sběrnici používanou u minipočítačů. Výhoda těchto sběrnic spočívá v tom, že časový konflikt na sběrnici není ničím, které jsou ke sběrnici připojeny. Tento konflikt je znázorněn na obr. 1 a je vidět, že je při něm pouze neplatná informace na sběrnici, neboť je na ní logický součet signálů A a B. Z obr. 1 je také vidět, proč tyto sběrnice přenášely obvykle data v negativní logice, to znamená, že úroveň 0 V na sběrnici znamenala log. 1. Na obr. 2 je princip časového multiplexu na sběrnici. Počítač vyrábí signály čtení, zápis a vzorkovací signál. Sám však používá tyto signály opačně! Je-li zápis, je ke sběrnici připojen výstup klopného obvodu C přes vyslač V₂. Informaci ze sběrnice převezme příjemce přes přijímač P₁, neboť signál vzorek projde při zápisu přes dekoder D₁. Při čtení je ke sběrnici připojen výstup klopného obvodu A přes vyslač V₁, informaci převezme počítač přes přijímač P₂, neboť signál vzorek projde přes dekoder D₂. Ve skutečném systému, kde je více než jeden zdroj a příjemce, stačí doplnit vys-



Obr. 2. Časový multiplex na datové sběrnici

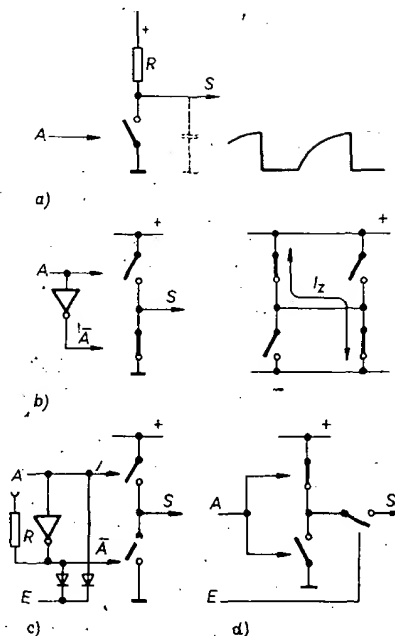
lače V_1 třetím vstupem, který povoluje jeho aktivaci pouze při jedné adrese, a dekodéry D_1 třetím, povolovacím vstupem, odvozeným z adresy.

Mikroprocesory používají třístavové sběrnice. Vysíláče pro třístavové sběrnice jsou zapojeny tak, že mají další vstup, který povoluje aktivaci výstupních tranzistorů vysíláče. Tím je umožněno řešit koncový stupeň jako dvojčinný (zrychlí se tím nabíjení kapacitní zátěže sběrnice), navíc ho lze zablokovat, aby se „nepřetahoval“ vysíláče od různých zdrojů informace.

Vývoj sběrnicových obvodů je na obr. 3. Na obr. 3a je vysíláč s otevřeným kolektorem. Na obr. 3b je běžný výstup hradla TTL, který nemůže být pro sběrnicové obvody použit pro proud I_z , který by protékal spínači různých vysíláčů, neboť ty mají pouze dva stavy a jeden ze spínačů je vždy sepnut. Na obr. 3c je princip třístavového hradla. Pokud je na vstupu E log. 1, pracuje hradlo jako běžný obvod TTL. Je-li na E log. 0, oba výstupní spínače jsou rozpojeny. Tím se odstraní možnost protékání proudu I_z . Pozor však na to, že při špatném časování signálu E může proud I_z protékat, čímž by byly krátkodobě, ale nesprávně přetěžovány výstupní spínače. Na obr. 3d je řešení třístavového výstupu v obvodu CMOS.

Zabývali jsme se datovými signály mikroprocesoru poněkud podrobněji, protože zde se dělá nejvíce chyb při aplikacích mikroprocesorových obvodů. Nesprávné časování dat na sběrnici se těžko určuje a je mu proto třeba věnovat při návrhu systému velkou pozornost.

Největší možnosti volby signálů dává poslední skupina, kterou jsou řídicí signály mikroprocesoru – je možné jednak ušetřit vývody, jednak si zvolit řídicí signály podle potřeby. Je zajímavé, že i když každý mikroprocesor plní prakticky stejné funkce, signály této skupiny se značně liší. Je to škoda, protože paměťové a periferní obvody mají poměrně standardní sběrnice a k vytvoření signálu pro jejich ovládání je třeba mnoha hradel a dekodérů. Navíc je třeba se „naučit“ časování a významy řídicích signálů. Díky právě tomu, že se řízení tolik liší, si konstruktér obvykle zvykne na jeden typ mikroprocesorů a nerad přechází na jiný. Proto se ještě dodnes používá v mnoha zařízeních 8080A. Stejně pravidlo platí totiž ještě o programech a vývojových a diagnostických pomůckách.



Obr. 3. Vývoj sběrnicových obvodů; a) vysíláč s otevřeným kolektorem, b) hradlo TTL s dvojčinným výstupem, c) třístavové hradlo TTL, d) třístavové hradlo CMOS

Mezi základní řídicí signály patří hodinové signály. Většina mikroprocesorů má vnitřní registry řešené jako dynamické a když ne, tak stejně potřebuje synchronizační hodinový signál, který určuje takt vykonávání instrukcí. Hodinové vstupy jsou řešeny buď pro připojení krystalu, nebo jako vstup TTL. Mikroprocesory, které vznikly v počátcích vývoje, mají obvykle vstupy, které vyžadují speciální úroveň a časování. Příkladem je opět 8080A, který má na „hodiny“ speciální požadavky. Na 8080A je však třeba dívat se jako na celek s jeho podpůrnými obvody 8224 a 8228, neboť zajišťují vše potřebné pro úroveň signálů hodin, jejich časování i pro synchronizaci ostatních řídicích signálů.

Dalším nezbytným signálem mikroprocesoru je vstup nulování, reset. Tento signál umožní uvést vnitřní obvody procesoru do známého definovaného stavu. Stejný signál se použije i pro nulování všech důležitých obvodů systému. Obvykle se nastaví programový čítač procesoru na nulu, vynuluje se klopný obvod povolující přerušení v procesoru a vynulují se registry přidavných zařízení a jejich obvody, žádající o přerušení.

Další skupina signálů souvisí s datovou a adresovou sběrnicí. Připojeným obvo-

dům je třeba sdělit, zda je na adresových linkách adresa paměti nebo přidavného zařízení. Dále je třeba vyslat signál o směru přenosu dat, tzn., chce-li procesor číst data – READ, nebo zapsat data – WRITE. Jejich význam pro obousměrný přenos dat vyplývá z obr. 2. Z obrázku je také vidět, že je nutné, aby existoval signál, který synchronizuje operaci zápisu. Při zápisu musí být na datových linkách ustálená a platná informace. Při operaci čtení počítá procesor s tím, že na základě adresy a signálu READ budou v určitém časovém okamžiku na sběrnici platná data a převezme je pomocí vnitřně generovaného vzorkovacího signálu. Řídící signály pro čtení a zápis se u mikroprocesorů značně liší, je proto třeba si pečlivě prostudovat časové diagramy a popisy jednotlivých signálů. Jde především o dvě základní pravidla: nikdy nesmí být ke sběrnici připojeny dva zdroje informace, a to ani na krátký okamžik, kdy se adresy zdrojů mění. Druhé pravidlo vyplývá z katalogových údajů procesoru a ostatních obvodů. Je třeba dodržet čas, kdy je informace na sběrnici platná, před vzorkovacím signálem tzv. časy setup time a čas, kdy informace zůstane platná po skončení vzorkovacího signálu, tzv. časy hold time.

Další skupina, obvykle dvojice signálů (READY, WAIT), souvisí s předešlou skupinou, která řídila čtení a zápis. Při použití pomalých pamětí nebo při zpoždění způsobeném dekodéry adres a řídicích signály by nebylo možné dodržet časování dat na sběrnici. Proto jsou mikroprocesory vybaveny vstupem, který zjišťuje, zvládnou-li přidavné obvody operaci v požadovaných časech. V případě, že by nebylo možné dodržet synchronizaci dat, je třeba oznámit pomocí tohoto vstupu procesoru žádost, aby počkal. Procesor si pak zařadí místo cyklu, který by přečetl informaci – tzv. čekací cyklus. Tento cyklus se periodicky opakuje a testuje se, zda již žádost o čekání zmizela. Jakmile žádost zmizí, procesor dokončí zápis nebo čtení. Aby byl vnějším obvodům znám stav procesoru, je obvykle vyveden výstup, který signalizuje, že procesor je ve stavu, kdy čeká na připravenost k dokončení zápisu nebo čtení. Tyto signály je možno použít i k jiným účelům.

Další skupina signálů patří k přerušovacímu systému. Co je přerušovací systém a k čemu se používá? Přerušovací systémy počítačů se rozvinuly až s použitím počítačů pro řízení rychlých procesů. Chce-li počítač vědět, je-li nějaké číslo záporné nebo kladné, zjistí si znaménkový bit testovací a skokovou instrukcí, neboli to udělá programem. Potřebuje-li počítač zjistit třeba stav tlačítek A, B a C, udělá to podobně. Vstupní instrukcí si přinese binární informaci o stavu tlačítka do svého pracovního registru a testovacími a skokovými instrukcemi zjistí stav těchto tlačítek. Co když je však tlačítek třeba dvacet a některá jsou ještě k tomu velmi důležitá, třeba POŽAR a POVO-DEN? Nebo co když potřebuje zjistit stav velice rychle? Například hlášení o tom, že začíná výpadek síťového napětí, mu nedává více času než 1 až 2 ms, aby si uložil potřebné informace do paměti, zálohované baterií. Není přece možné, aby při výpočtu nějaké úlohy se periodicky ptal na stav důležitých tlačítek a signálů. To by toho moc neudělal. Proto bylo zavedeno tzv. přerušení, doslova přerušení právě probíhajícího programu. Přerušení je vlastně vnucená instrukce skoku, která se automaticky provede po skončení právě dokončené instrukce. Aby nebyl program přerušován, musí-li počítač dokončit již započatý úkol, třeba rychlý přenos dat

z disku, existuje v přerušovací systém klopny obvod, který blokuje (neboli maskuje) přerušení. Některé mikroprocesory pak mají dva vstupy pro žádost o přerušení, maskovatelný a nemaskovatelný. Na první se připojují žádosti sice důležité, ale takové, které je možno alespoň na čas ignorovat. Na druhý pak žádosti, které nelze ignorovat, třeba zmíněné hlášení o výpadku síťového napájení.

Je-li přerušovací systém počítače použit, zjednoduší se obsluha důležitých stavů a hlášení, která do systému přicházejí od řízeného procesoru nebo od přidavných zařízení. Přerušovací systém však přinesl do počítačů i řadu problémů, které bylo nutné vyřešit. Přerušíme-li právě probíhající program, musíme pak vědět, kde máme pokračovat. Je také nutné uložit si dočasně obsahy registrů a obnovit je, než budeme pokračovat v přerušeném výpočtu. Bylo by také nutné zavést prioritu žádostí o přerušení, která umožní počítači ignorovat méně důležité žádosti, obsluhuje-li právě tu důležitější. Máme-li pouze jeden vstup žádosti o přerušení do procesoru a dvacet tlačítek, neboli dvacet důležitých hlášení, moc by nám přerušení nepomohlo. Museli bychom po přerušení stejně zjišťovat, kterým tlačítkem se o přerušení žádalo: bylo by třeba dvacet dotazů na stav jednotlivých tlačítek. Moderní přerušovací systémy používají tzv. *vektor*, což jsou data, přenášená po datové sběrnici a blíže specifikující místo, které žádalo o přerušení a tím vlastně i to, kam má směřovat nucený skok v programu, aby bylo žádající místo co nejdříve obsluženo podprogramem uloženým v paměti. Shrňme-li požadavky na signály, patří k přerušovacímu systému, pak je to především žádost o přerušení, která do procesoru vstupuje a dále signál o potvrzení přerušení, který obvykle zajišťuje synchronizaci přenosu vektoru po datové sběrnici z místa žádosti do procesoru. U sběrnic spojujících periferní obvody bývají ještě signály, které zajišťují prioritu žádosti a synchronizaci systému přerušení.

Poslední skupina signálů zajišťuje tzv. přímý přístup k paměti. Co to je přímý přístup k paměti? Je-li k počítači připojeno nějaké rychlé přidavné zařízení (disková paměť, rychlý převodník A/D atd.), bylo by zdůluhavé přenášet data po sběrnici ze zařízení do registrů počítače a pak je zapsat instrukcí zápisu do paměti. Není-li třeba, aby počítač data průběžně zpracovával nebo kontroloval, nebo kdyby to ani časově nestačil, je možné ho z přenosu dat vypustit. Data je možno přímo ze zařízení poslat na datovou sběrnici a vnutit paměti signál *zápis*. Stejně tak při opakném přenosu třeba z paměti na disk je možno vnutit paměti signál *čtení* a data, která vyšle paměť na datovou sběrnici, zapsat přímo do registru v řadiči disku. Místo programového přenosu, kdy jde všechno přes počítač, probíhá pak přímý přenos mezi zařízením a pamětí a rychlost tohoto přenosu je dána rychlostí paměti. Není však možné jen tak, z ničeho nic, posílat na sběrnici data a řidiči signály *čtení* a *zápis* a navíc, jak jste jistě postřehli, je nutné generovat i adresu paměti, kam nebo odkud data posíláme. Proto mají mikroprocesory vstupy a výstupy pro signály, které umožní „zapůjčit“ sběrnici jinému, jednoduššímu, avšak rychlejšímu zařízení, kterému říkáme řadič přenosu DMA. Tento řadič pak požádá mikroprocesor o sběrnici pomocí signálu *žádost o sběrnici* a mikroprocesor mu potvrdí, že mu sběrnici zapůjčuje, signálem *potvrzení žádosti*. Předtím, než mikroprocesor žádost potvrdí, odpojí si své sběrnice (adresu, data

i řízení) tím, že je uvede do třetího stavu. Je obvyklé, že procesor potvrzuje žádost o zapůjčení sběrnice i uprostřed instrukce, jen dokončí svůj právě probíhající cyklus přenosu pro sběrnici. Tím je dána i rychlá reakce na požadavek přenosu DMA, která ještě zvětšuje možnosti využití velké rychlosti tohoto způsobu přenosu dat. Celá operace je pak řízena řadičem DMA, který generuje postupně adresy a řidičí signály a hlídá i počet přenesených dat. Po skončení přenosu pak řadič DMA zruší žádost o zapůjčení sběrnice a mikroprocesor zruší signál o potvrzení zapůjčení a může pracovat dále.

Je samozřejmé, že není možno vyčerpat všechny možné kombinace signálů u různých mikroprocesorů a proto je důležité pochopit principy spolupráce počítače s okolím a umět si odvodit, proč a jak jednotlivé signály fungují.

Sběrnice mikropočítačových systémů

Doplníme-li mikroprocesor o paměti, obvody vstupních a výstupních zařízení a napájecí zdroj, dostaneme mikropočítač. Podaří-li se nám vše umístit na jednu desku s plošnými spoji, pak takovému mikropočítači říkáme jednodeskový. Obvykle však žádný mikropočítač dlouho jednodeskový nežijí. Další aplikace potřebují speciální obvody pro vstup a výstup, větší paměť a postupně další a další desky pro rozšíření systému. Protože jednou z největších výhod mikropočítačů je jednoduchost při výrobě systémů pro různé aplikace, je nutné co nejméně zjednodušit propojení desek systému. V prvních mikropočítačových systémech se desky spojovaly technikou ovíjených vodičů a proto bylo možné přidávat a ubírat signály podle potřeby. Dnešní mikropočítačové systémy používají pro propojení jednotlivých desek maticí desku s plošnými spoji (back plane). Aby byl motiv těchto desek jednoduchý a univerzální, používá se paralelní spojení odpovídajících špiček konektorů. Je samozřejmé, že tímto způsobem lze dosáhnout značné ekonomie výroby „kabeláže“, ale současně je omezen počet signálů, které mohou desky spojit. Je to tedy podobná situace jako u vlastního mikroprocesoru, u něhož jsme omezeni počtem vývodů pouzdra. Jak uvidíme dále, je problém spojování nejen desek, ale i různých zařízení, vstupních čidel a výstupních akčních členů atd. jedním z největších problémů počítačové techniky. Proto je vždy snaha redukovat počet vodičů potřebných k přenosu informací na minimum, zejména s rostoucí délkou potřebných kabelů.

A tím jsme vlastně u vzniku systémových sběrnic neboli sběrnic propojujících jednotlivé bloky systému. Desky mikropočítačových systémů jsou spojeny sběrnici, která jednak rozvádí napájecí napětí a jednak přenáší adresové, datové a řidičí signály; jejich význam je stejný jako u mikroprocesorů, protože sběrnice je vlastně prodloužením vývodů mikroprocesoru k ostatním blokům mikropočítačového systému.

Jednodeskový mikropočítač má obvykle paměti RAM a PROM a základní obvody vstupu a výstupu na desce. Mikroprocesor a obvody umístěné na desce se propojují samozřejmě také sběrnici, které říkáme vnitřní sběrnice desky. Systémová sběrnice připojená na konektor desky se od vnitřní sběrnice liší tím, že je „výkonová“, to znamená, že obvody vysílající signály na sběrnici mají schopnost budit delší vodiče i větší zátěže. Systémová

sběrnice musí být navržena tak, aby bylo možno rozšířit systém o další desky. Praxe je taková, že je-li některá sběrnice populární (třeba S 100), snaží se jak amatéři, tak elektronické firmy vyrábět desky pro tuto sběrnici. Aby bylo možné vymyslet jednoznačné obvody pro komunikaci se sběrnici, je nutné, aby každá sběrnice byla co nejpřesněji definována (viz úvod této kapitoly). Vytvořit sběrnici a zpracovat pro ni definici není jednoduché. Příkladem může být sběrnice S 100, která není zcela jednoznačně definována (nebo signál AACK sběrnice MULTIBUS, který musel být z původní definice vypuštěn, protože vznikaly problémy při jeho aplikaci).

Jako příklad jednoduché, dobře navržené sběrnice uvedu STD BUS. Tato sběrnice patří do skupiny sběrnic pro malé systémy. Sběrnice MULTIBUS, která se používá i u většiny československých mikropočítačových systémů, patří naopak do skupiny sběrnic pro větší systémy, které se dnes svým výkonem blíží minipočítačům.

Sběrnice STD BUS

STD BUS je sběrnice vyvinutá firmou Pro-Log Corporation. Sběrnice je definována pro přímý konektor a obsahuje 56 signálů. Standardní desky mají šířku 4,5" a výšku 6,5", s konektorem na kratší straně. Právě pro tento konektor, který má neobvyklou rozteč kontaktů 0,125", je tato sběrnice u nás zatím nepoužitelná. V USA se sběrnice používá zejména pro aplikaci mikropočítačů v řízení procesů a strojů a v systémech měřicích ústředí. Desky pro sběrnici STD vyrábí mnoho firem, z nichž nejznámější jsou Pro-Log a Mostek. Sběrnice je definována s ohledem na mikroprocesor Z80 a pro něj také vycházejí nejjednodušší obvody styku se sběrnici. Obsazení špiček konektoru sběrnice STD BUS je v tab. 1. Z ní je vidět, že sběrnice má čtyři skupiny signálů. Napájecí napětí jsou rozdělena do dvou podskupin. První obsahuje napájení pro logické obvody +5 V, zem a -5 V a druhá napájení pro analogové obvody +12 V, -12 V a zem.

Datová část sběrnice (vývody 7 až 14) obsahuje osm obousměrných datových linek. Data se přenášejí v pozitivní logice třístavovými vysílací.

Adresová část sběrnice (vývody 15 až 30) obsahuje 16 adresových linek. Spodních 8 adresových linek je vyvedeno na liché špičky a horních 8 na sudé špičky konektoru. Stejně jako u datových linek je požadováno připojení přes třístavové vysíláče.

Řidičí část sběrnice je nejobsáhlejší. Špičky 31 a 32 přenášejí signály zápis a čtení. Všechny signály řidičí části s výjimkou prioritních vstupů a výstupů jsou přenášeny v negativní logice, tj. aktivní v nule. Špičky 33 a 34 přenášejí signály o platnosti adresy. Na desce jsou tyto signály logicky znásobeny se signály čtení a zápisu a získají se signály pro čtení a zápis pro přidavná zařízení a nebo čtení a zápis pro paměti. Toto členění základních řidičích signálů vychází z uspořádání procesoru Z80 a pro procesor 8080A není vhodné.

Špičky 35 a 36 slouží k rozšíření počtu adresovaných zařízení a k rozšíření paměti. Těchto signálů lze různě využít, jedna možnost spočívá v jejich využití ve funkci dalšího adresového vodiče. Na desce pro-

Tab. 1. Zapojení konektoru sběrnice STD BUS

| Skupina | Strana součástek | | | | Strana spojů | | | |
|-------------------|------------------|----------|----|--|--------------|----------|----|--|
| | Číslo | Název | AÚ | Funkce | Číslo | Název | AÚ | Funkce |
| Napájení | 1 | +5 V | | napájení | 2 | +5 V | | napájení |
| | 3 | GND | | | 4 | GND | | |
| | 5 | -5 V | | | 6 | -5 V | | |
| Datová sběrnice | 7 | D3 | H | data | 8 | D7 | H | data |
| | 9 | D2 | H | | 10 | D6 | H | |
| | 11 | D1 | H | | 12 | D5 | H | |
| | 13 | D0 | H | | 14 | D4 | H | |
| Adresová sběrnice | 15 | A7 | H | adresa (spodní linky) | 16 | A15 | H | adresa (horní linky) |
| | 17 | A6 | H | | 18 | A14 | H | |
| | 19 | A5 | H | | 20 | A13 | H | |
| | 21 | A4 | H | | 22 | A12 | H | |
| | 23 | A3 | H | | 24 | A11 | H | |
| | 25 | A2 | H | | 26 | A10 | H | |
| | 27 | A1 | H | | 28 | A9 | H | |
| | 29 | A0 | H | | 30 | A8 | H | |
| Řídící sběrnice | 31 | WR | L | zapis adresa periferie rozšíření adresy refres stav procesoru zapůjčení sběrnice potvrzení přerušení žádost o čekání nulování systému hodiny procesoru výstup prior. řetězce | 32 | RD | L | čtení adresa paměti rozšíření adresy synchronizace stav procesoru žádost o sběrnici požadavek na přerušení požadavek na nemask. přerušení tlačítko nulování pomocné hodiny vstup prior. řetězce |
| | 33 | IORQ | L | | 34 | MEMRQ | L | |
| | 35 | IOEXP | L | | 36 | MEMEX | L | |
| | 37 | REFRESH | L | | 38 | MC SYNC | L | |
| | 39 | STATUS 1 | L | | 40 | STATUS 0 | L | |
| | 41 | BUSAK | L | | 42 | BUSRQ | L | |
| | 43 | INTAK | L | | 44 | INTRQ | L | |
| | 45 | WAITRQ | L | | 46 | NMIHQ | L | |
| | 47 | SYSRESET | L | | 48 | PB RESET | L | |
| | 49 | CLOCK | L | | 50 | CNTRL | L | |
| | 51 | PCO | H | | 52 | PCI | H | |
| Napájení | 53 | AUX GND | | pomocné napájení +12/+15 V | 54 | AUX GND | | pomocné napájení -12 V/-15 V |
| | 55 | AUX +V | | | 56 | AUX -V | | |

Pozn. AÚ = aktivní úroveň

Tab. 2. Zapojení konektoru K1 sběrnice ARB-1

| Č.1 | Signál | Název | Typ | Č. | Signál | Název | Typ |
|------------------|--------|----------------------|------------------------|-------------|----------------|--|-----|
| 1 | RTL | tlačítko RESET | INP | 2 | HOLD | žádost o DMA | IN |
| 3 | RDY | READY | INP | 4 | STSTB | vzorek STATUSU | OUT |
| 5 | HLDA | potvrzení pro DMA | OUT | 6 | M1 | příznak cyklu M1 | OUT |
| 7 | INTE | přerušení povoleno | OUT | 8 | RES | nulování | OUT |
| 9 | MR | čtení z paměti | OUT | 10 | IOR | čtení z portu | OUT |
| 11 | MW | zápis do paměti | OUT | 12 | IOW | zápis do portu | OUT |
| 13 | AEN | povolení adres | OUT | 14 | DEN | povolení dat | INP |
| 15 | +5 V | napájení | NAP | 16 | +5 V | napájení | NAP |
| 17 | +5 V | | NAP | 18 | +5 V | | NAP |
| 19 | 0 V | | NAP | 20 | 0 V | | NAP |
| 21 | | zem | | 22 | | zem | |
| 23 | A9 | | | 24 | A15 | | |
| 25 | A11 | | | 26 | A14 | | |
| 27 | A13 | | | 28 | A12 | | |
| 29 | A10 | adresa | OUT | 30 | A8 | adresa | OUT |
| 31 | D4 | | | 32 | D3 | | |
| 33 | D6 | | | 34 | D5 | | |
| 35 | D2 | | | 36 | D7 | | |
| 37 | D0 | data | BD | 38 | D1 | data | BD |
| 39 | A1 | | | 40 | A0 | | |
| 41 | A3 | | | 42 | A2 | | |
| 43 | A5 | | | 44 | A4 | | |
| 45 | A7 | adresa | OUT | 46 | A6 | adresa | OUT |
| 47 | | | | 48 | | | |
| 49 | | | | 50 | | | |
| 51 | +12 V | | | 52 | +12 V | | |
| 53 | 0 V | napájení | NAP | 54 | 0 V | napájení | NAP |
| 55 | -5 V | | | 56 | -5 V | | |
| 57 | -12 V | | | 58 | -12 V | | |
| 59 | INT1 | | | 60 | Φ ₂ | | |
| 61 | INTA | žádost o přerušení 1 | INP | 62 | INT0 | žádost přerušení 0 | INP |
| | | potvrzení přerušení | OUT | | | | |
| Číslo konektoru: | | K1 | Konektor: Protikus: | TX 518 6212 | | INP – vstupní BD – obousměrný OUT – výstup NAP – napájení | |
| Klíčování: | | F3 | | TY 517 6211 | | | |

cesoru mohou být signály generovány třeba z výstupního portu a na deskách paměti a přídavných zařízeních mohou vybrat druhou sadu registrů nebo paměťových bloků.

Špička 37 přenáší informaci o tom, že na adresové části sběrnice je obcerstvovací adresa pro dynamické paměti. Procesorové desky používající Z80 nepotřebují pro obcerstvení žádné přídavné obvody. Desky využívající jiných procesorů musí mít přídavné obvody pro řízení funkcí obcerstvení a čítač adres (refresh, address counter).

Signály na špičkách 42 a 41 patří do skupiny signálů umožňujících přenos DMA a několik procesorové aplikace.

Špičky 43 a 44 slouží pro obsluhu přerušovacího systému. I když jsou tyto signály zvoleny opět pro procesor Z80, dovolují realizovat přerušovací systém pro procesor 8080A. Špička 46 přenáší žádost o nemaskovatelné přerušení a je využívána pro přerušení od výpadku napájení. Přerušovací systém sběrnice je řetězový. Špičky 51 a 52 tvoří prioritní řetězec a sběrníkový vodič spojuje pouze špičky sousedních konektorů. Rozpojení sběrnice umožňuje realizovat na desce přerušovací systém odpovídající periferním obvodům systému Z80.

Na špičkách 38, 39 a 40 jsou signály identifikující strojový cyklus procesoru. Tyto signály jsou závislé na použitém procesoru.

Špička 45 slouží k požadavku na zařazení čekacích cyklů a to na dobu, po níž je signál v nule. Signál umožňuje pomalejší paměť a obvodům spolupráci s rychlejšími procesory.

Špička 47 přenáší nulovací signál všem obvodům připojeným ke sběrnici, které potřebují nastavit definovaný počáteční stav. Špička 48 slouží pro tlačítko RESET nebo obecně pro signál, který požaduje generaci signálu na špičce 47.

Špička 49 přenáší zesílený hodinový signál procesoru a špička 50 pomocný hodinový signál, který je obvykle podílem hodinového signálu. Špičku 50 je možno použít také pro hodiny reálného času, představované nejčastěji signálem odvozeným z napětí síťového kmitočtu.

Sběrnice STD BUS má ze všech průmyslových sběrnic nejmenší rozměry desky. I když používá přímý konektor, je oblíbená zejména pro průmyslové aplikace. Spolehlivost přímého konektoru je určena hlavně kvalitou konektoru a tolerancí tloušťky desky. U nás není možno úzké tolerance dodržet a proto je všeobecně považován přímý konektor za nespolehlivý. Malý rozměr desky umožňuje tuhou a jednoduchou mechanickou konstrukci jedno a dobré chlazení. Podélný tvar desek s vedením po delší straně přispívá také k vyšší spolehlivosti. V neposlední řadě je malý formát vhodný pro optimální modularitu řídicích mikropočítačových systémů. Současným světovým trendem v konstrukci řídicích systémů jsou malé desky, a proto je sběrnice STD stále perspektivní.

Signály sběrnice ARB-1

Sběrnice ARB-1 je deska s oboustrannými plošnými spoji, která je osazena konektory FRB, K1 až K8. Použité konektory mají 62 kontaktů ve dvou řadách a rozteč kontaktů 2,5 mm. Vývody konektorů jsou číslovány od 1 do 62 tak, že jedna řada má lichá čísla 1 až 61, druhá sudá 2 až 62. Nejvhodnější typ konektoru pro desku sběrnice je TX 5186212, který má vývody pro zapojení do desky s plošnými spoji.

Deska s plošnými spoji slouží jednak pro mechanické upevnění konektorů a jednak pro vzájemné propojení vývodů konektorů K1 až K8. Konektor K1, který je první zprava, je určen pouze pro desku procesoru (do něj se zasouvá deska procesoru JPR-1) a proto má odlišné klíčovací naváděcí kolíčky vzhledem ke konektorům K2 až K8. Vyhradit speciální pozici pro desku procesoru u sběrnice ARB-1 bylo nutné proto, že na desce sběrnice jsou kromě konektorů ještě čtyři obvody MH3216, které zesilují adresové signály vycházející z desky procesoru. Z konstrukčních důvodů je proto mezi konektory K1 a K2 dvojnásobná mezera (40 mm) než mezi konektory K2 až K8 (20 mm).

Jednotlivým vodičům, propojujícím konektory ať už přímo, nebo přes zesilovače, přísluší určité číslo vývodu konektoru, označení signálu zkratkou a název signálu. Pro konektor K1 platí tab. 2 a pro konektory K2 až K8 tab. 3. Při prvním pohledu na obě tabulky vidíme, že se liší zejména v tom, že některé signály jsou pro procesor výstupy a pro ostatní desky vstupy a naopak. Další rozdíl je v signálech označených jako S1 až S6, které z konstrukčních důvodů propojují jen pozice konektorů K2 až K8. Tyto signály jsou určeny jako rezerva a nedoporučují je zatím používat.

Před tím, než popíšeme jednotlivé signály sběrnice ARB-1, bych chtěl zdůraznit, že prakticky všechny signály jsou vlastně „prodlouženými vývody“ základního „osmdesátkového“ systému, tvořeného trojicí obvodů: 8080A (mikroprocesor), 8224 (hodiny) a 8228 (systémový řídicí obvod). Sběrnice ARB-1 je proto „šitá na míru“ pro mikroprocesor 8080A. Tím se zjednoduší problém definice této sběrnice. Až na prakticky zanedbatelná zpoždění, vzniklá zesilováním některých signálů, a na „aktivní úroveň“ jednotlivých signálů je časování sběrnice shodné se systémem mikroprocesoru 8080A. Aktivní úroveň logického signálu nazýváme ten stav, kdy signál dělá to, co dělat má. Tak například aktivní úroveň signálu \overline{MR} je „nula“ (L), neboť právě při této úrovni se čte z paměti. Aktivní úroveň signálu STSTB je „jednička“ (H), protože právě při úrovni tohoto signálu +3 V se vzorkuje status z datových linek obvodu 8080A. Aktivní úroveň řídicích signálů na sběrnicích jsou obvykle L, aby v klidu byl signál v úrovni H, neboť rušivá napětí u obvodů s logikou TTL musí být mnohem větší pro H, než pro L (neboť rozhodovací úroveň je 1,4 V a úroveň L je 0,4 V, kdežto úroveň H je větší než 3 V). U datových a adresových linek lze těžko hovořit o aktivní úrovni, neboť u nich je stejně významná úroveň L jako H. Navíc platí adresa i data vždy po dobu aktivní úrovně řídicího signálu (\overline{MR} , \overline{MW} , \overline{INTA} apod.), takže krátké rušení na těchto linkách nebývá tak nebezpečné. Aktivní úroveň L jsou u všech signálů označeny inverzí zkratky signálu.

Popis signálů sběrnice ARB-1

Signál \overline{RTL} (1) slouží k připojení tlačítka RESET. Tlačítko, zapojené mezi zem a \overline{RTL} , musí vybit kondenzátor 20 μ F přes odpor 100 Ω na napětí 0,8 V. Při použití kontaktního tlačítka je doba navybíjení vždy dostatečná. Použijeme-li však pro nulování místo tlačítka třeba výstup obvodu 7406, je třeba pamatovat na to, že šifka nulovacího impulsu by měla být alespoň několik ms. Nulovací člen \overline{RC} je na desce procesoru a je připojen na vstup RESIN obvodu 8224. Signál \overline{RTL} je vyveden jako jediný řídicí signál z desky ARB-1 na

Tab. 3. Zapojení konektorů K2 až K8 sběrnice ARB-1

| Č. | Signál | Název | Typ | Č. | Signál | Název | Typ |
|----|-------------------|---------------------|-----|----|--------------------|-----------------------|-----|
| 1 | \overline{RTL} | tlačítko RESET | OUT | 2 | \overline{HOLD} | žádost o DMA | OUT |
| 3 | \overline{RDY} | READY | OUT | 4 | \overline{STSTB} | vzorek STATUSU | INP |
| 5 | \overline{HLDA} | potvrzení pro DMA | INP | 6 | $\overline{M1}$ | příznak cyklu M1 | INP |
| 7 | \overline{INTE} | přerušení povoleno | INP | 8 | \overline{RES} | nulování | INP |
| 9 | \overline{MR} | čtení z paměti | BD | 10 | \overline{IOR} | čtení z portu | BD |
| 11 | \overline{MW} | zápis do paměti | BD | 12 | \overline{IOW} | zápis do portu | BD |
| 13 | \overline{AEN} | povolení adres | INP | 14 | \overline{DEN} | povolení dat | - |
| 15 | +5 V | napájení | NAP | 16 | +5 V | napájení | NAP |
| 17 | +5 V | | NAP | 18 | +5 V | | NAP |
| 19 | 0 V | | NAP | 20 | 0 V | | NAP |
| 21 | S1 | rezerva 1 | - | 22 | S2 | rezerva 2 | - |
| 23 | A9 | adresa | BD | 24 | A15 | adresa | BD |
| 25 | A11 | | BD | 26 | A14 | | BD |
| 27 | A13 | | BD | 28 | A12 | | BD |
| 29 | A10 | data | BD | 30 | A8 | data | BD |
| 31 | D4 | | BD | 32 | D3 | | BD |
| 33 | D6 | | BD | 34 | D5 | | BD |
| 35 | D2 | adresa | BD | 36 | D7 | adresa | BD |
| 37 | D0 | | BD | 38 | D1 | | BD |
| 39 | A1 | | BD | 40 | A0 | | BD |
| 41 | A3 | rezerva 3 | BD | 42 | A2 | rezerva 4 | BD |
| 43 | A5 | | BD | 44 | A4 | | BD |
| 45 | A7 | | BD | 46 | A6 | | BD |
| 47 | S3 | rezerva 5 | - | 48 | S4 | rezerva 6 | - |
| 49 | S5 | napájení | NAP | 50 | S6 | napájení | NAP |
| 51 | +12 V | | NAP | 52 | +12 V | | NAP |
| 53 | 0 V | | NAP | 54 | 0 V | | NAP |
| 55 | -5 V | zem | NAP | 56 | -5 V | zem | NAP |
| 57 | -12 V | | NAP | 58 | -12 V | | NAP |
| 59 | $\overline{INT1}$ | žádost o přerušení | OUT | 60 | Φ_2 | hodiny Φ_2 - TTL | INP |
| 61 | \overline{INTA} | potvrzení přerušení | INP | 62 | $\overline{INT0}$ | žádost o přerušení 0 | OUT |

Číslo konektoru:
Klíčování:

K2 až K8
C6

Konektor
Protikus

TX 518 62 22
TY 517 62 11

INP – vstup
BD – obousměrný
OUT – výstup
NAP – napájení

Tab. 4. Požadavky na budiče na deskách systému JPR-1

| Č. | Název. | Typ a proud | Obvod | Č. | Název | Typ a proud | Obvod |
|----------------------|--------------------------------------|--|------------------------|----------------------|--------------------------------------|------------------------------|-----------------------------|
| 1 3 | \overline{RTL} \overline{RDY} | kontakt nebo výk. OK 5 mA, 1k na +5 V, OK | 7406 7405, 03 | 2 4 | \overline{HOLD} | 5 mA, 1k na +5 V, OK | 7405, 03 |
| 9 11 | \overline{MR} \overline{MW} | } třístavový výstup 10 mA | 74126 (3216) | 10 12 | \overline{IOR} \overline{IOW} | } třístavový výstup 10 mA | 74126 (3216) |
| 23 25 27 29 | A9 A11 A13 A10 | | } jako \overline{MR} | 3216 | 24 26 28 30 | | A15 A14 A12 A8 |
| 31 33 35 37 | D4 D6 D2 D0 | } obousměrné, třístavové 10 mA | | | 32 34 36 38 | D3 D5 D7 D1 | } obousměrné, třístavové |
| 39 41 43 45 | A1 A3 A5 A7 | | } jako \overline{MR} | 40 42 44 46 | A0 A2 A4 A6 | } jako \overline{MR} | |
| 59 61 | $\overline{INT1}$ | 5 mA, 1k na +5 V, OK | | 7405, 03 | 62 | | $\overline{INT0}$ |

svorku, ke které je možno připojit externí tlačítko RESET.

Signál \overline{HOLD} (2) má aktivní úroveň L. Inverzní signál je veden na vstup \overline{HOLD} procesoru 8080A. Signálem předáváme procesoru žádost o DMA, neboli o zapůjčení adresové, datové a řídicí sběrnice.

Signál \overline{HLDA} (5) má aktivní úroveň L a je inverzní signálu \overline{HLDA} procesoru 8080A. Signál oznamuje, že procesor žádost o DMA akceptoval, a že tedy půjčuje sběrnicí jiné desce systému.

Signál \overline{RDY} (3) je přímo vstupním signálem \overline{RDYIN} obvodu hodin 8224. Na desce procesoru je pouze zakončen odporem 1 k Ω na +5 V. Tento signál je

obvodem 8224 synchronizován s hodinovým signálem procesoru a poslán již jako synchronní obvodu 8080A pod označením READY.

Signál \overline{STSTB} (5) je invertovaným výstupním signálem z vývodu 7 obvodu 8224. Tento synchronizační signál slouží ke vzorkování stavového slova, které se posílá po vnitřní datové sběrnicí na začátku každého strojového cyklu 8080A.

Signál $\overline{M1}$ (6) je vlastně invertovaným bitem D5 vnitřní datové sběrnice systému

Tab. 5. Maximální zátěže budičů desky JPR-1 a sběrnice ARB-1

| Č. | Název | Typ a proud | Obvod | Č. | Název | Typ a proud | Obvod |
|----|-------|----------------------|---------|----|----------------|----------------------|--------|
| 5 | HDA | 9 zátěží TTL, 14 mA | 7404 | 4 | STSTB | 10 zátěží TTL, 16 mA | 7404 |
| 7 | INTE | 10 zátěží TTL, 16 mA | 7404 | 6 | MI | 10 zátěží TTL, 16 mA | 7404 |
| 9 | MR | 4 zátěže TTL, 7 mA | 7404 | 8 | RES | 9 zátěží TTL, 14 mA | 7404 |
| 11 | MW | | (3205) | 10 | IOR | | 7404 |
| | | | | 12 | IOW | 6 zátěží TTL, 10 mA | (3205) |
| 15 | +5 V | Ø 0,75 A/desku | | 16 | +5 V | | |
| 17 | +5 V | | | 18 | +5 V | | |
| 23 | A9 | | 7404 | 24 | A15 | | 7404 |
| 25 | A11 | 10 zátěží TTL, 16 mA | (3205) | 26 | A14 | 10 zátěží TTL, 16 mA | (3205) |
| 27 | A13 | | (3212) | 28 | A12 | | (3212) |
| 29 | A10 | | | 30 | A8 | | |
| 31 | D4 | 8228 10 mA | 3216 | 32 | D3 | stejně jako D0 | |
| 33 | D6 | 2708 0,6 mA | | 34 | D5 | | |
| 35 | D2 | 2114 1,1 mA | 4x 3216 | 36 | D7 | | |
| 37 | D0 | 2716 1,1 mA | 4x 3216 | 38 | D1 | | |
| 39 | A1 | | 7404 | 40 | A0 | 10 zátěží TTL, 16 mA | 7404 |
| 41 | A3 | | (3205) | 42 | A2 | | (3205) |
| 43 | A5 | | (3212) | 44 | A4 | | (3212) |
| 45 | A7 | | | 46 | A6 | | |
| 51 | +12 V | Ø 0,25 A/desku | | 52 | +12 V | | |
| 55 | -5 V | | | 56 | -5 V | | |
| 57 | -12 V | Ø 0,06 A/desku | | 58 | -12 V | | |
| 59 | | | | 60 | Φ ₂ | 9 zátěží TTL, 14 mA | 74132 |
| 61 | INTA | 6 zátěží TTL, 10 mA | | 62 | | | |

Pozn. 1: Maximální zátěž dat závisí na osazení desky JPR-1 paměťovými obvody. Povolené zátěže datových linek jsou proto rozepsány podle obvodů použitých na JPR-1.

Pozn. 2: Obvody 3205, 3212, 3216 mají vstupní proud 6x menší než TTL. Proto jsou výhodnější než 7404 apod.

8080A. Tento bit má ve stavovém slově funkci příznaku cyklu FETCH (čtení instrukce z paměti). Ve spojení se signálem STSTB je pak možno identifikovat cykl FETCH pro diagnostické a ladicí účely.

Signál RES (8) je invertovaným výstupním signálem RESET obvodu 8224. Signál RESET je také veden do procesoru 8080A, v němž vyvolá funkci nulování systému. Signálu RES se využívá na sběrnici k nu-

lování nebo blokování funkcí na ostatních deskách systému JPR-1.

Signály INTO, INT1 (62), (59) patří do skupiny signálů systému přerušení. Jsou to vlastně vstupní signály obvodu 3214, který zajišťuje funkci přerušovacího systému JPR-1. Signál INTE (7) je invertovaným výstupním signálem INTE procesoru 8080A. Tento signál přenáší informaci o povolení nebo zákazu přerušení. Signál INTA (61) je přímo výstupním signálem INTA obvodu 8228 a potvrzuje, že procesor 8080A akceptoval žádost o přerušení.

Signály MR (9) a MW (11) jsou přímo výstupními signály obvodu 8228. MR je signál pro časování čtení z paměti a MW pro časování zápisu do paměti.

Signály IOR (10) a IOW (12) jsou opět výstupními signály obvodu 8228. IOR je signál pro časování čtení ze vstupních portů a IOW je signál pro časování zápisu do výstupních portů.

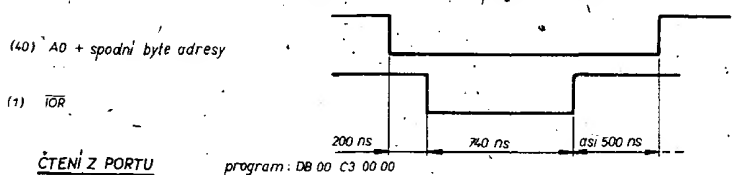
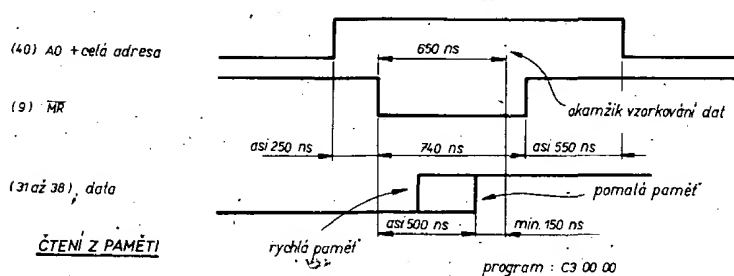
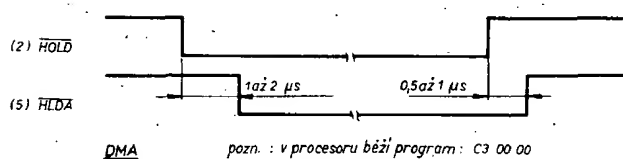
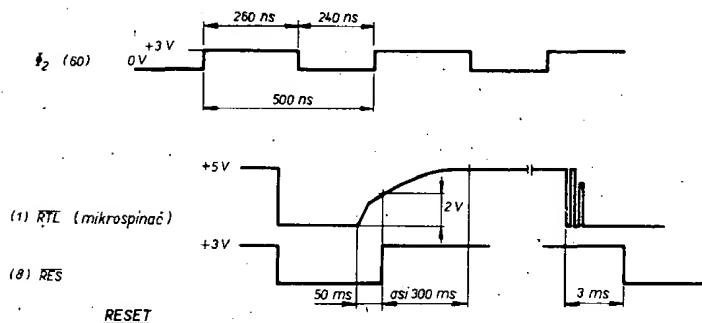
Signál Φ₂ je výstup Φ₂ (TTL) obvodu 8224. Tento signál má kmitočet rovný 1/9 kmitočtu krystalu a střídá 5:9. Signál slouží k synchronizaci nebo jako zdroj hodinového signálu pro desky systému JPR-1.

Signály dat D0 až D7 (31 až 38) jsou datové vstupy/výstupy obvodu 8228.

Signály spodního bytu adresy A0 až A7 (39 až 46) slouží k adresaci paměti a portů.

Signály horního bytu adresy A8 až A15 (23 až 30) slouží k adresaci paměti.

Signál DEN (14) je vstupním signálem obvodu 8228, který se nazývá BUSEN. Tento signál je na desce JPR-1 uzemněn přes odpor 100 Ω a slouží na sběrnici pouze k diagnostickým účelům! Vnucení úrovně H na tento vstup umožňuje uvést datové a řídicí výstupy obvodu 8228 do třetího stavu.



(40) A0 + spodní byte adresy

(12) IOW

(31 až 38) data

ZÁPIS DO PORTU

(40) A0 + spodní byte adresy

ZÁPIS DO PAMĚTI

(4) STSTB

(7) INTE

POVOLENÍ PŘERUŠENÍ

(7) INTE = INT

(61) INTA

8080A

spojeno

INTA

INT

8228

data

INTA

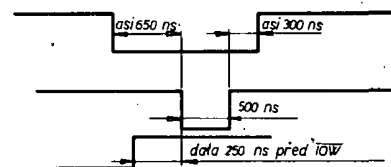
3212

E16

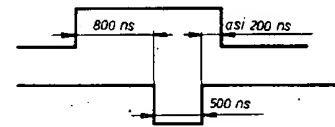
D7

INTA

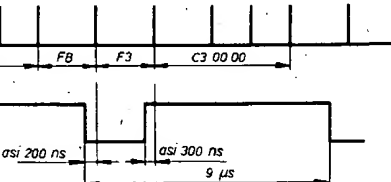
PŘERUŠENÍ



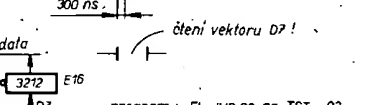
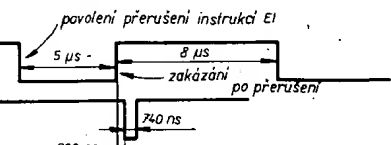
program: D3 00 C3 00 00



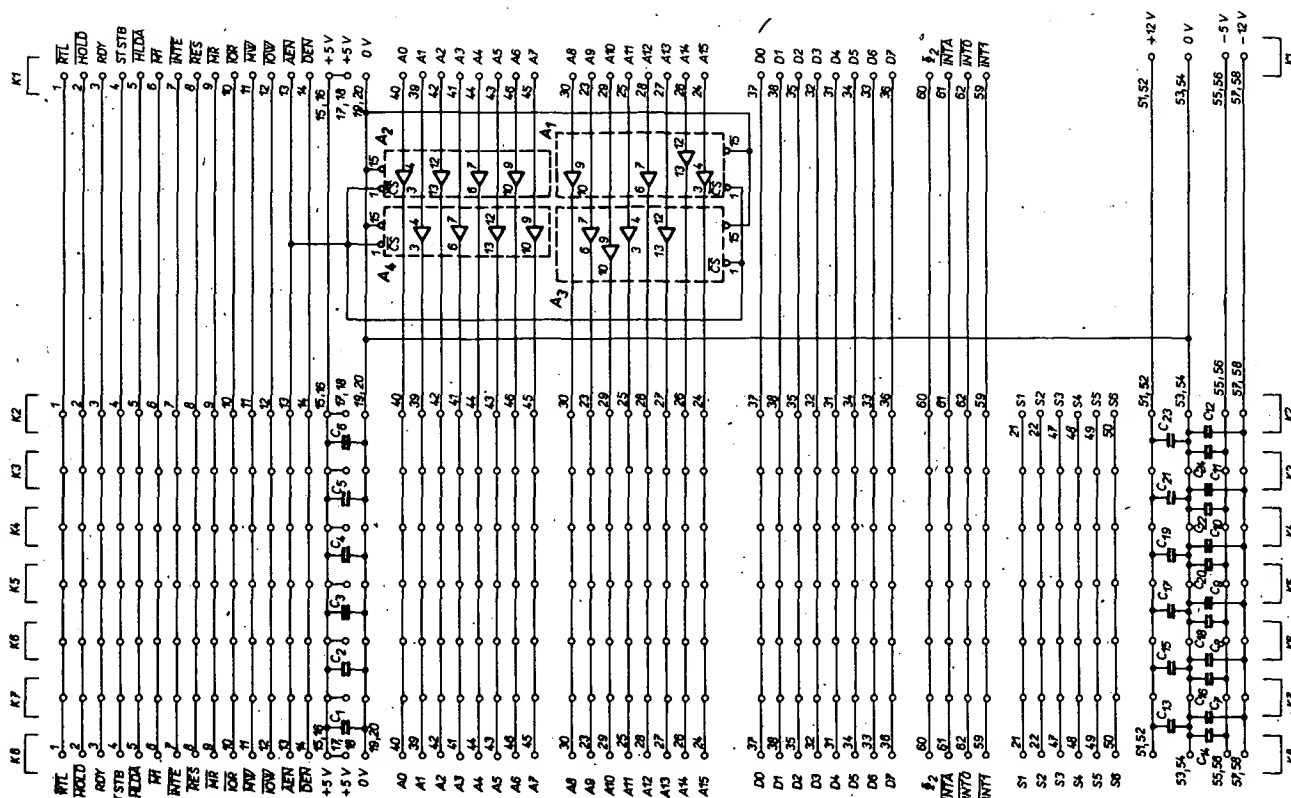
program: RST0 - C7



program: FB, F3, C3 00 00



POZOR! U přípravky TST-02 (diodová ROM) jsou všechny RST zpět na 0000!

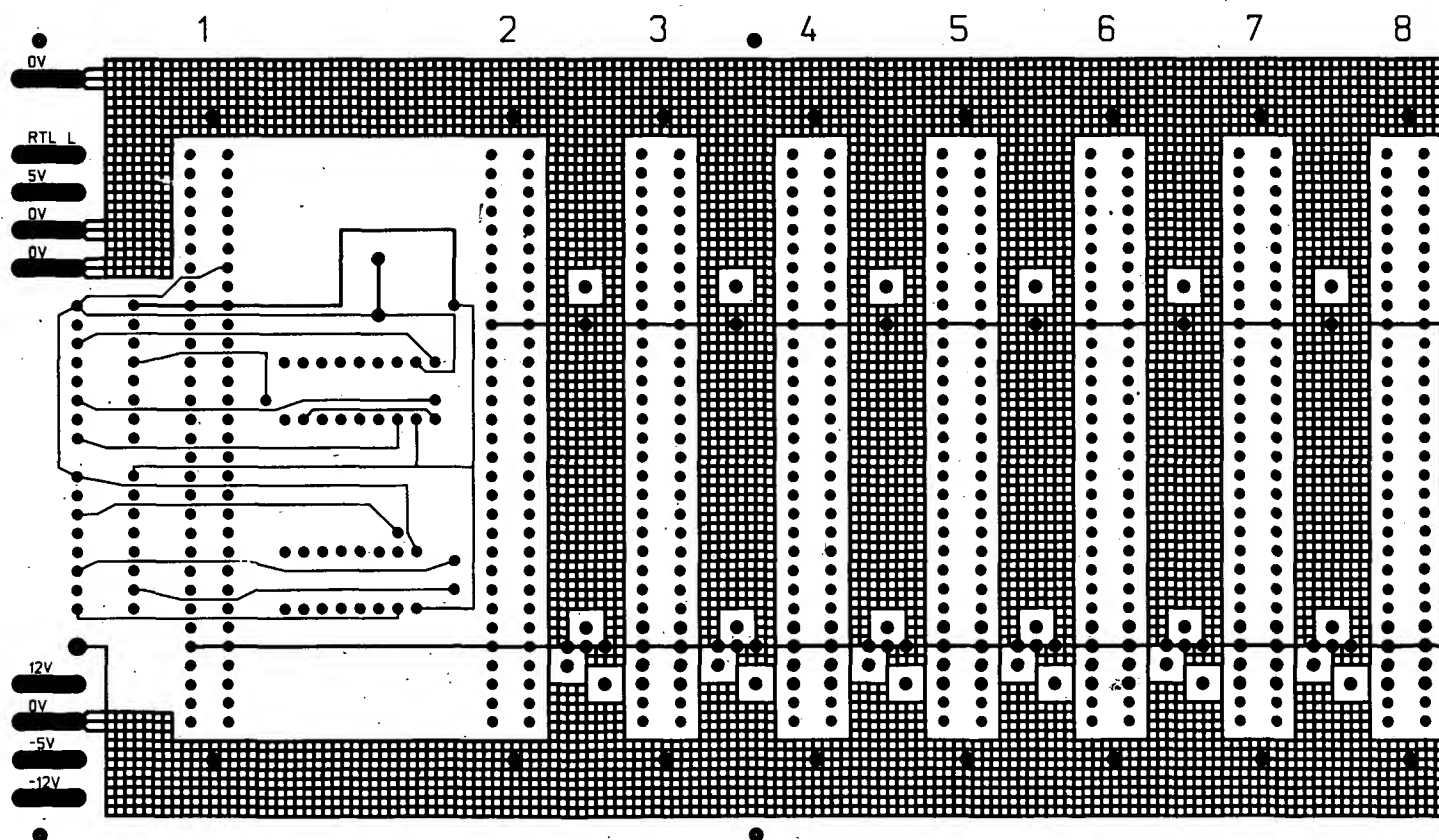


Obr. 5. Schéma sběrnice ARB-1

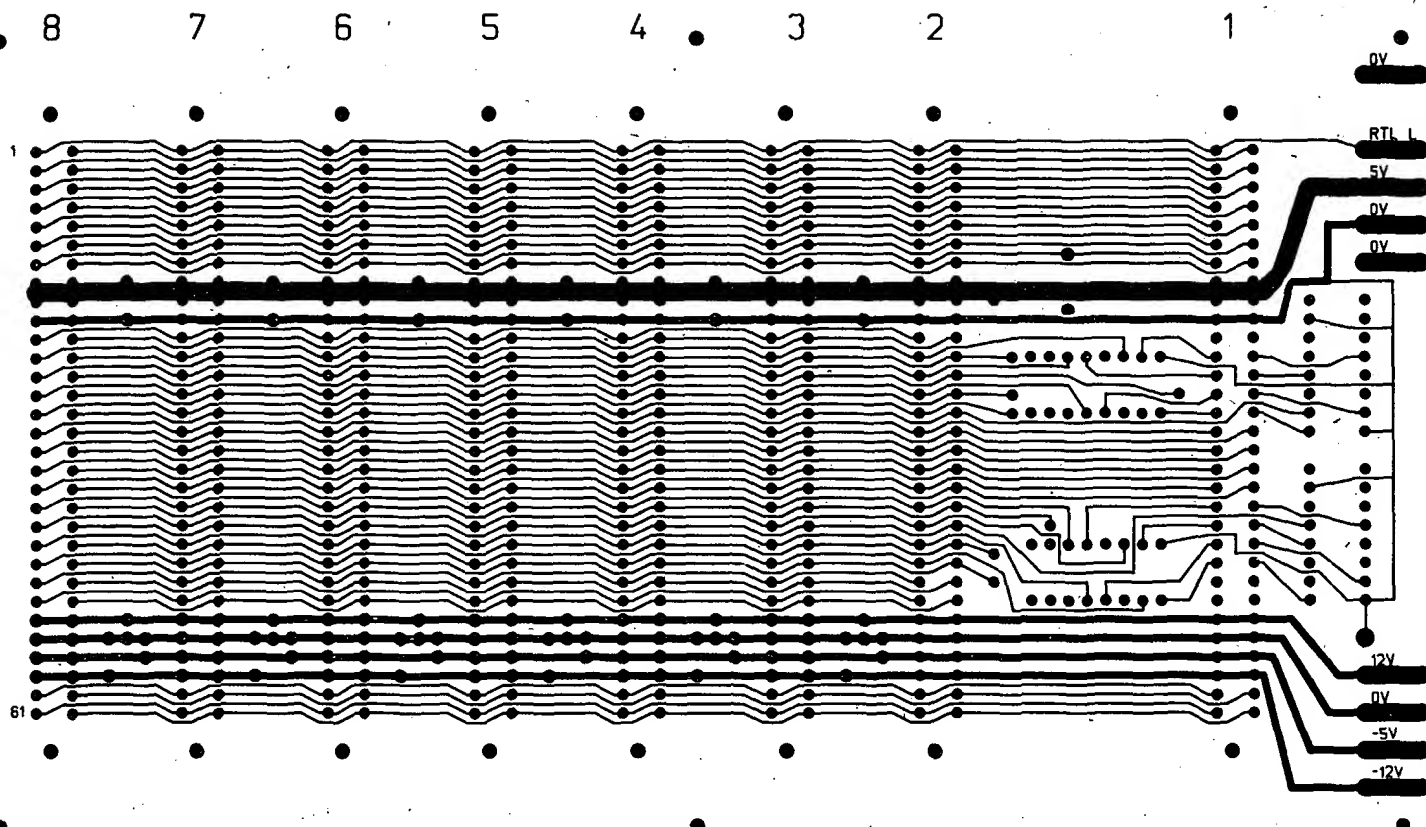
Signál AEN (13) je vlastně invertovaným signálem sběrnice HLDA. Signál AEN slouží k uvedení zesilovačů adres 3216 na desce sběrnice ARB-1 do třetího stavu, je-li potvrzena žádost o DMA. Na ostatních deskách systému je možno tento signál využít jako inverzi HLDA.

Signály napájení desek sběrnice jsou rozvedeny z přírodních svorek desky sběrnice ARB-1 na jednotlivé konektory K1 až K8. Zapojení je v tab. 2 a platí pro celý systém JPR-1. Na desce ARB-1 je napájení průběžně blokováno kondenzátory C₁ až C₂₄.

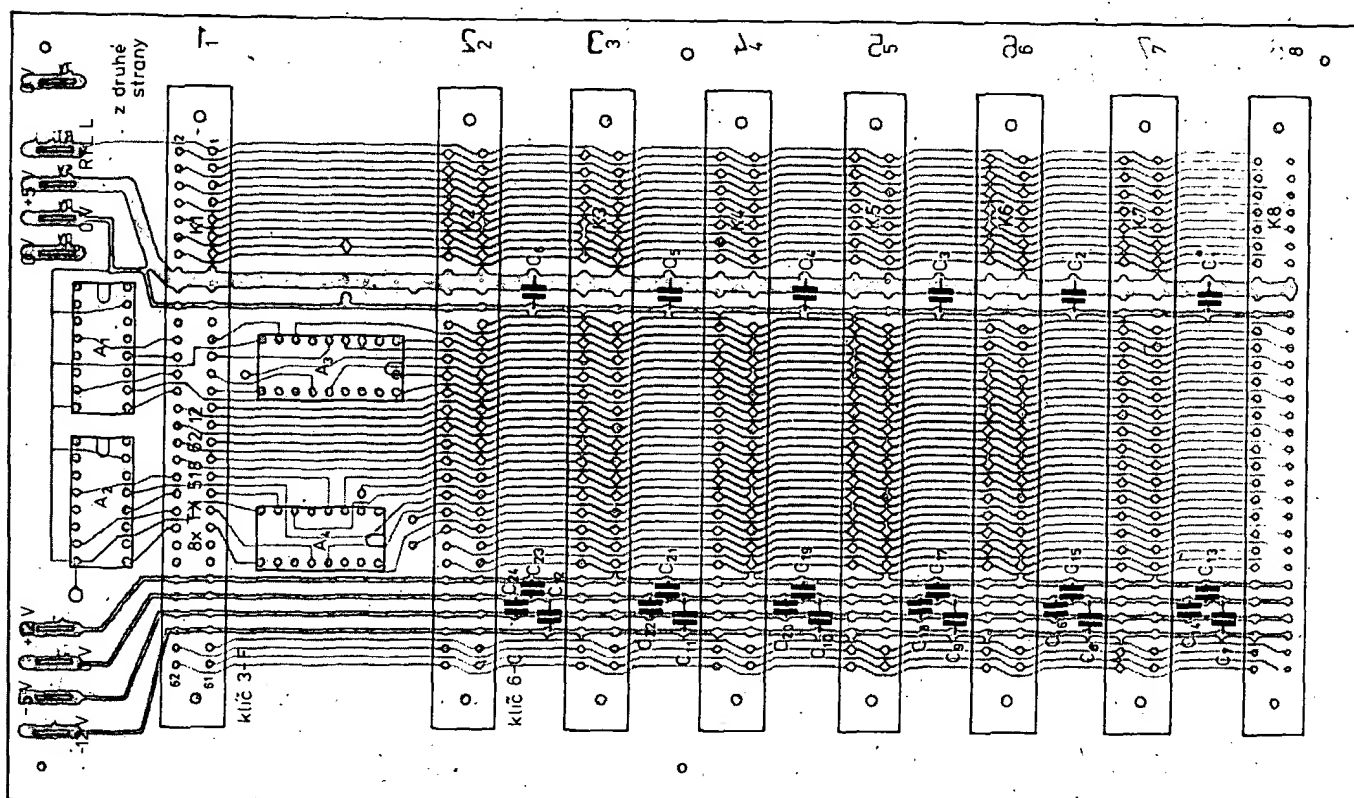
Pro buzení a zatěžování signálů sběrnice ARB-1 platí informace v tab. 4 a 5, v nichž je pro každý vstup nebo výstup sběrnice uveden i vhodný obvod, který může se signálem spolupracovat. Na obr. 4 jsou časové vazby mezi jednotlivými signály sběrnice měřené



Obr. 6. Deska s plošnými spoji sběrnice ARB-1, strana součástek



Obr. 7. Deska s plošnými spoji ARB-1, spodní strana



Obr. 8. Rozložení součástek na desce sběrnice ARB-1

s procesorem JPR-1 s kmitočtem hodinového signálu (krystalu) 18 MHz. Tyto průběhy jsou dobrým pomocníkem při práci se systémem JPR-1.

Konstrukce sběrnice ARB-1

Schéma desky je na obr. 5. Jak je vidět, zapojení je skutečně jednoduché. Deska s plošnými spoji má rozměr 200 × 115 mm a je oboustranná; není nutné, aby měla prokovené díry. Blokovací kondenzátory C_1 až C_{24} je lépe dát z opačné strany desky, než jsou konektory. Při zasouvání desek by ohnutý kondenzátor mohl vadit. Napájecí napětí jsou přivedena přes konektorové svorky, používané v automobilech. Svorky jsou vyvedeny směrem dozadu a svorky pro tlačítko RESET směrem dopředu. Z přední strany desky ARB-1 jsou pak připojeny vodiče, vedoucí napájení na systémový panel, z něhož vede napájení do klávesnice. Na obr. 6 a 7 je deska s plošnými spoji sběrnice ARB-1. Na obr. 8 je rozložení součástek.

Oživení desky

Po zapojení konektorů a ostatních součástek se vyplácí změřit možné zkratky mezi vodiči a špičkami, na nichž jsme pájeli. Jakost pájení pak zkontrolujeme lupou. Potom zkontrolujeme logickou sondou funkci zesilovačů adres 3216. Pomůžeme si postupným „zemněním“ adresových vývodů konektoru K1 a měříme na konektorech K2 až K8. Dále zkontrolujeme funkci signálu AEN (13). Ten musí být při měření zesilovačů uzemněn! Odpojíme-li ho, musí se všech 16 adresových zesilovačů uvést do třetího stavu. Dále změříme při připojení zdroje špičky rozvodu napájecího napětí. Měříme i na sousedních špičkách, abychom vyloučili zkratky. Do konektorů sběrnice budeme zasouvat drahé desky a proto se vyplácí vše měřit a kontrolovat velmi pozorně. V žádném případě nezkoušíme sběrnici tím, že do ní zasuneme procesor JPR-1 a desky pamětí!

Seznam součástek pro desku sběrnice ARB-1

Pasívní součástky

C_1 až C_{12} 22 nF, TK 783
 C_{13} až C_{24} 15 nF, TK 783

Integrované obvody

A1, A2, A3, A4 MH3216

Ostatní

K1 až K8 konektor FRB, TX 518 62 12
 deska s plošnými spoji sběrnice ARB-1
 automobilový konektor, 9 ks

DESKA PROCESORU JPR-1

Deska JPR-1 vznikla na základě potřeby mít jednoduchý mikroprocesorový systém, který by byl schopen nahradit pevně zapojenou logiku. I když je potřeba „něco mít“ dlouhodobá, obvykle se přimějeme k tomu, abychom „to“ udělali, až „to“ potřebujeme nutně. K takové situaci došlo, když jsem potřeboval elektroniku k elektrickému psacímu stroji CONSUL 256, který se dodává bez řídicí elektroniky. Má bezkontaktní klávesnici a vyžaduje poměrně složité časování pro spínání ovládacích magnetů. Stroj je vybaven zpětnovazebními čidly, které indikují realizaci požadovaných funkcí. Magnety ty-

pových pák jsou zapojeny do matice, která neodpovídá přesně kódu ASCII.

Nejprve jsem navrhl a postavil řídicí systém pro tento stroj jako sekvenční automat, pracující podle obsahu uloženého v bipolárních pamětech PROM MH74S287 (256 × 4). Takto navržený systém fungoval, ale měl minimální šanci na další využití.

Po zvážení práce, kterou by bylo nutné vynaložit na konstrukční dohotovení automatu, jsem se rozhodl, že použiji raději mikroprocesor 8080A. V té době jsem měl možnost seznámit se s dokumentací jednoho zahraničního řídicího systému. Tento systém řídil jinou, velice často používanou periférii počítačů, digitizér pro snímání souřadnic X, Y z nakreslené předlohy. Systém byl postaven na bázi základních součástek mikroprocesorového systému Intel 8080 – prostudování této dokumentace vlastně rozhodlo o vzniku JPR-1.

Procesor byl nejprve navržen jako jednodeskový a potom postupně předělán tak, že může sloužit jako základní deska mikropočítače. Hlavním cílem vývoje bylo, aby vznikla deska s plošnými spoji, kterou by bylo možno osadit podle požadovaných počtů vstupů a výstupů základními obvody, které se u nás vyrábějí nebo budou vyrábět. Většina jednodeskových mikropočítačů je složitá, protože se počítá, že i jednodeskový mikropočítač bude později rozšířen o další desky pamětí a přídatných zařízení. JPR-1 byl navržen trochu jinak. Na desce procesoru nejsou výkonové zesilovače dat a adres, které obvykle oddělují vnitřní sběrnice jednodeskového mikropočítače od sběrnice systému, proto se na ni podařilo umístit poměrně velkou kapacitu pamětí a velký počet vstupů a výstupů. Navržené řešení je výhodné, použijeme-li JPR-1 v aplikacích, v nichž všechno „uřídí“ sám; chceme-li ho však použít jako základ několika-deskového mikropočítačového systému, musíme použít sběrnici ARB-1 se zesilovači adresových signálů a navíc musíme pamatovat na to, že paměťové obvody na desce JPR-1 jsou vlastně „pověšeny“ přímo na datovou sběrnici. Uvedené vlastnosti a celková jednoduchost dokazují výhodnost systému, který se podle procesoru nazývá JPR-1.

Konstrukce systému JPR-1

Stanovení rozměru desky se spoji je vždy základním krokem vývoje nového počítačového nebo mikropočítačového systému. Klíčem k jakékoli normalizaci rozměrů desek (MULTIBUS, CAMAC, STD BUS) je konektor. Formát desky JPR-1 (140 × 150 mm) vznikl postupně,

při vývoji desky procesoru a celého systému. Nejprve byla deska o něco menší a postupně se zvětšovala, aby bylo možno použít pro výstup na přídatná zařízení konektory FRB, které jsou sice pro tyto účely moc velké, ale jiné nemáme.

Na konečný formát desky se hodně vejde, dobře se dají vést spoje ke konektorům a systém dvou konektorů se 30 vývody pro připojování přídatných zařízení se dobře osvědčil u systému JPR-12. Konektor FRB, použitý u sběrnice, není sice příliš vhodný, protože má malou rozteč vývodů, ale nebylo z čeho vybírat. Konektory pro sběrnice se obvykle dělají s větší roztečí (0,15" a 0,156" a 0,125"), aby bylo více místa na protažení spojů (jak na desce sběrnice, tak na deskách systému).

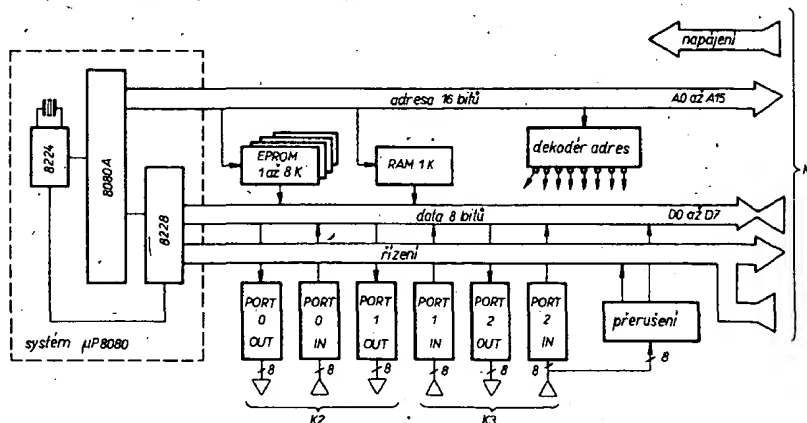
Systém JPR-1 je určen pro aplikace v oblasti řízení strojů, přídatných zařízení počítačů a technologických zařízení. Jde tedy o systém profesionální, a proto jsou všechny desky s plošnými spoji navrženy tak, že vyžadují prokovené díry. Při jejich návrhu však bylo pamatováno na to, aby počet průchodů z jedné strany na druhou byl minimální, a aby spoje nebyly příliš „husté“. Vzorky všech desek jsem postavil na deskách bez prokovení, je to sice pracné, ale jde to.

Blokové schéma desky procesoru JPR-1

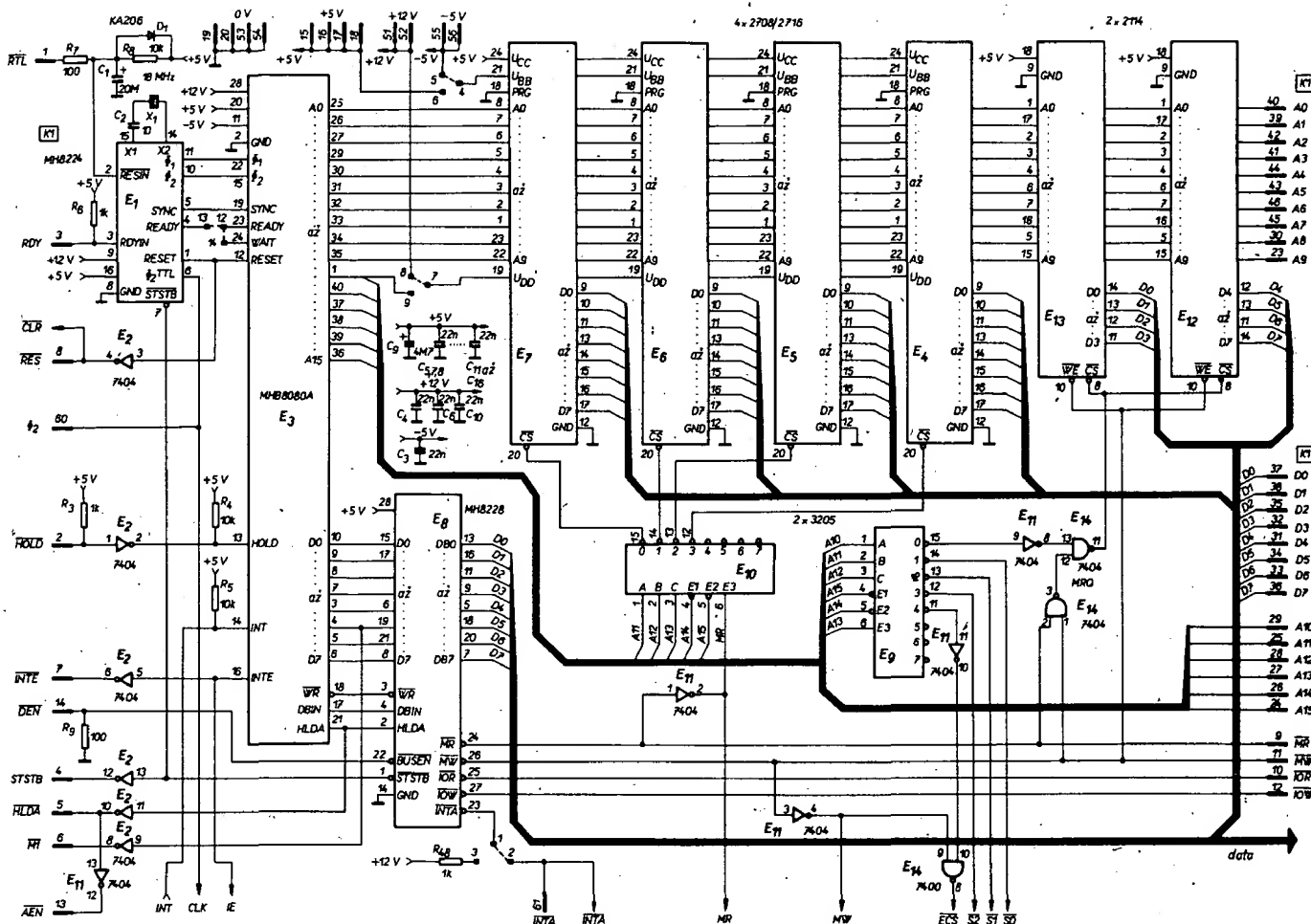
Blokové schéma desky je na obr. 1, připomíná základní obrázky z knížek a článků, zabývajících se mikroprocesory. Jednoduchý mikropočítač s mikroprocesorem 8080A však jinak zapojit nelze. Při návrhu jednoduchého mikropočítače je nutné vyřešit hlavně tři problémy:

1. Způsob adresace pamětí a portů.
 2. Vybrat takové zapojení a obvody, aby nebyly překročeny zátěže výstupů jednotlivých obvodů.
 3. Zpracovat konstrukci desky, to znamená navrhnout desku s plošnými spoji a zhotovit podklady pro jeho výrobu.
- Z vlastní zkušenosti mohu potvrdit, že bod 3 je 80 % práce při vzniku nového mikropočítače. Ostatní je již možné nechat na programátorovi, který si musí umět s danou konfigurací mikropočítače nějak poradit. Při návrhu mikropočítačového systému je navíc důležitá i určitá počáteční koncepce, neboť desky systému vznikají postupně a jakékoli zásadní změny se pak těžko promítají do již hotových věcí.

Podívejme se nyní na blokové schéma. Konektor K1 je vlastně konektor sběrnice desky JPR-1. Bude-li JPR-1 použit jako jednodeskový mikropočítač, použijeme



Obr. 1. Blokové schéma desky JPR-1



Obr. 2. Schéma desky JPR-1, procesor + paměti

sběrnici pro oživení desky a pro případné ladění programů pomocí krokování programu na přípravku TST-03. Bude-li JPR-1 základem několikadeskového systému, bude konektor K1 vlastně systémovou sběrnici.

Konektory K2 a K3 jsou určeny pro připojení přídatných zařízení. Na tyto konektory je vyvedeno 24 vstupů a 24 výstupů portů mikropočítače. Protože konektory mají málo špiček, je vstupní port, označený pořadovým číslem 3, použit současně jako vstup osmi přerušovaných vstupů.

Blokové schéma nám také prozradí, že deska JPR-1 má kapacitu paměti EPROM 1 až 8K a paměti RAM 1K. Jako paměti EPROM lze použít buď obvody 2708 (1K), nebo 2716 (2K). Jako paměti RAM jsou použity dva obvody 2114. Máme-li dán počet vstupů a výstupů a kapacitu paměti, pak jediné místo, kde můžeme ušetřit na obvodech, je dekodér adresy. Dekodování adresy u JPR-1 odpovídá jednoduchosti minimálního mikropočítače s mikroprocesorem 8080A. Proto jsou adresy paměti EPROM voleny po 2K, aby nebylo nutné přepínat vstupy dekodéru při použití paměti různých kapacit. Dekodér adres je z obvodů 3205 a ty mají pouze 6 vstupů. Protó adresace paměti a portů vychází pouze z adresových bitů A10 až A15. Porty jsou adresovány jako paměť a jeden port pak „zabírá“ vlastně tisíc adres paměťového prostoru 64K. Kdybychom chtěli adresovat porty jako I/O, přibyl by nejméně dva obvody, a ty by se již na desku nevešly. Myslím si však, že „nějakého toho kila“ není škoda, protože minipoch-

tač JPR-12 měl maximální kapacitu paměti 4K a přesto se pro něj našlo dost aplikací.

Na blokovém schématu je také vidět, že JPR-1 má sběrnici rozdělenou na čtyři obvyklé části: napájení, adresa, data a řízení. První tři části jsou dány použitým mikroprocesorem. Řídící část sběrnice byla volena tak, aby umožnila systém oživit a dále rozšiřovat.

Schéma zapojení desky JPR-1

Schéma je pro přehlednost rozděleno na dvě části. Na obr. 2 jsou obvody mikroprocesoru, paměti a dekodér adres. Na obr. 3 jsou vstupní a výstupní porty a přerušovací systém mikropočítače. Signály, které jsou zakončeny šipkami, přicházejí z jedné části schématu na druhou. Základem celé desky je známá trojice obvodů 8224, 8228 a 8080A.

Obvod 8224 je generátor hodinových signálů pro procesor 8080A. Obdobně jako většina prvních mikroprocesorů potřebuje i 8080A hodiny dvofázové s větším rozkmitem signálu a speciálním časováním.

Zapojení vývodů obvodu 8224 je na obr. 4. Obvod vyrábí signály Φ_1 a Φ_2 ze signálu o kmitočtu krystalu, připojeného k vývodům X1 a X2. Průběhy fází hodin jsou odvozeny ze stavů děličky devíti, která je uvnitř obvodu. Po dobu stavu 1 a 2 je generován signál Φ_1 . Po dobu stavů 3, 4, 5, 6 a 7 je generován signál Φ_2 . Po dobu stavu 9 je generován signál STSTB, ale jen tehdy, přichází-li z obvodu 8080A signál SYNC. Jako vedlejší produkty vnitřních obvodů generátoru hodin jsou vyvedeny signály OSC – signál s kmitočtem krystalového oscilátoru o úrovni TTL, a signál

Φ_2 (TTL) – signál Φ_2 o úrovni TTL. Tyto signály slouží pro všeobecné využití. U JPR-1 je využit pouze Φ_2 (TTL).

Dále obvod 8224 tvaruje signál RESIN a následně synchronizuje vzniklý průběh s hodinami Φ_2 . Vznikne tak signál RESET, který je nulovacím signálem pro obvod 8080A. Obdobně, jenže bez tvarování, je synchronizován vstup RDYIN a je generován signál READY pro obvod 8080A. Oba signály jsou synchronizovány na náběžnou hranu Φ_2 .

Na obr. 5 je zapojení vývodů 8228. D0 až D7 představuje obousměrnou datovou sběrnici mezi obvody 8080A a 8228. Této sběrnici můžeme říkat vnitřní datová sběrnice, po ní se přenáší jak data, tak stavový byte, kterému se říká STATUS. Obvod 8080A má totiž pouze 40 vývodů, což je málo pro všechny signály potřebné k řízení připojených pamětí, portů a pro obsluhu přerušování. Obvod 8228 má za úkol „vyrobit“ potřebné řídicí signály na základě převzatého stavového byte.

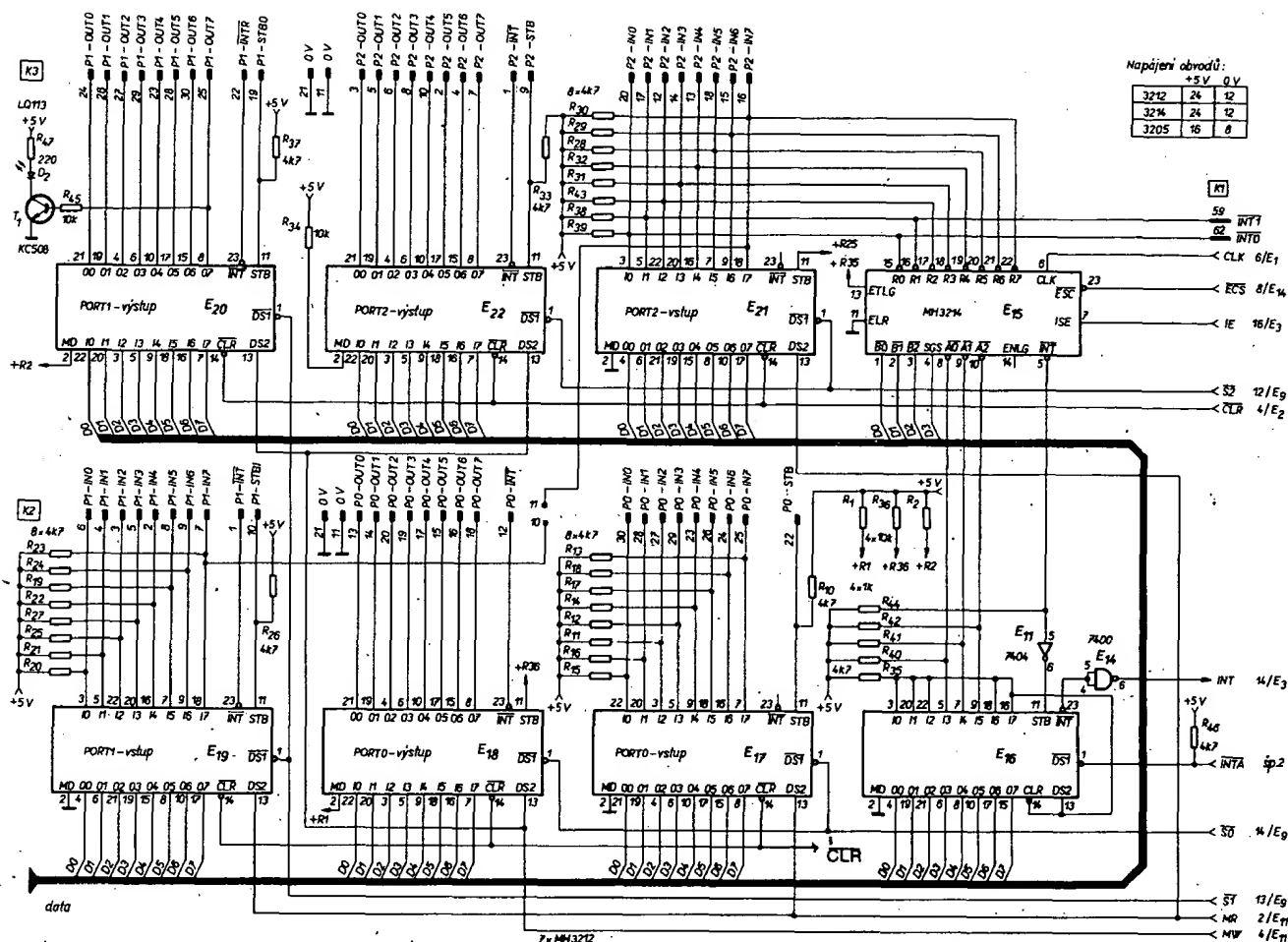
DB0 až DB7 představují „výkonovou“ sběrnici, po níž se přenáší obousměrně data z připojených pamětí a portů. Této sběrnici můžeme říkat systémovou.

Vstup STSTB již známe z obvodu 8224. Tento vstup, aktivní v nule, oznamuje obvodu 8228, že je na vnitřní datové sběrnici stavový byte.

Vstupy WR a DBIN jsou signály, které přicházejí z obvodu 8080A a řídí směr a časování přenosu dat po vnitřní datové sběrnici.

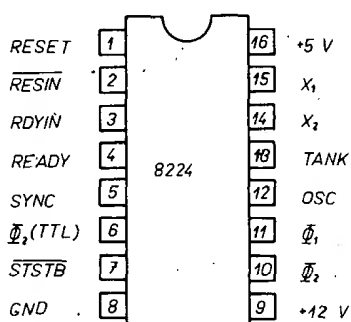
K systémové sběrnici patří také řídicí signály, aktivní při úrovni L a výkonově zesílené:

MEMR vzorkovací signál pro čtení z paměti,
MEMW vzorkovací signál pro zápis do paměti,

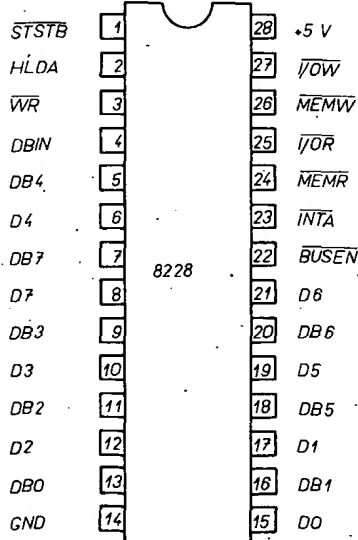


Obr. 3. Schéma desky JPR-1, porty

Tranzistor T_1 s diodou D_2 (LED) umožňuje indikovat stav výstupu 7 portu P1 pro diagnostické účely (test procesoru)



Obr. 4. Zapojení vývodů 8224



Obr. 5. Zapojení vývodů 8228

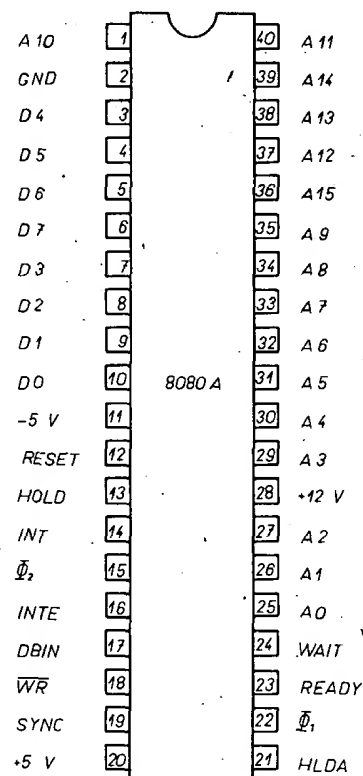
I/\overline{OR} vzorkovací signál pro čtení z portů,
 I/\overline{OW} vzorkovací signál pro zápis do portů,
 $INTA$ potvrzení přerušení a vzorkovací signál pro čtení instrukce RST nebo CALL, které u systému 8080A nahrazují obvyklý vektor přerušení.

Posledními dvěma vstupy obvodu 8228 jsou HLDA a BUSEN, které umožňují uvést výstupy systémové sběrnice do třetího stavu. HLDA se používá při přenosu dat DMA a BUSEN k řízení výstupů při ožiování a testování systému.

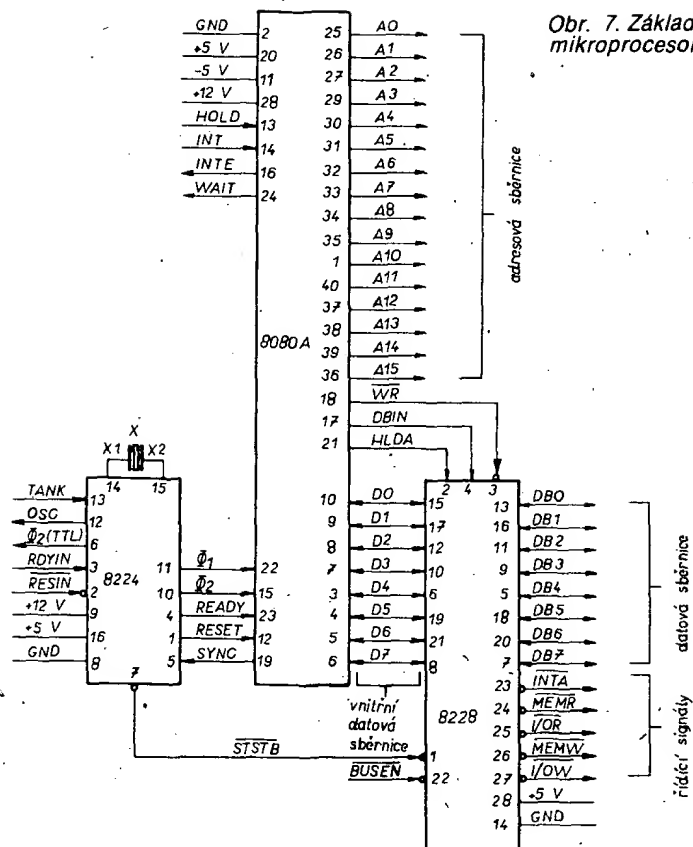
Mikroprocesor 8080A je posledním z trojice obvodů. Zapojení jeho vývodů je na obr. 6. Výstupy A0 až A15 tvoří adresovou sběrnici. Tato sběrnice slouží pro adresaci paměti nebo portů. Pro adresaci paměti se používá všech šestnácti adres a pro porty jen spodních osm. U systému JPR-1 není adresace portů zatím použita a paměti i vstupy a výstupy se adresují stejně, pomocí plné 16bitové adresy.

Obousměrné datové signály D0 až D7 již známe, protože jdou přímo na obvod 8228. V systému JPR-1 se navíc používá bit D5, který přenáší ve stavovém byte informaci o tom, že následující cyklus bude čtení instrukce z paměti (cyklus FETCH). Informace o tomto cyklu, který se nazývá také M1, můžeme pak využít pro krokování programu po instrukcích nebo i při návrhu dalších desek systému (obnovování – refresh dynamických pamětí, dekódování instrukcí atd.).

Vývody Φ_1 , Φ_2 , READY, RESET a SYNC jdou na obvod 8224 a vývody WR, DBIN a HLDA na obvod 8228.



Obr. 6. Zapojení vývodů 8080A



Obr. 7. Základní zapojení mikroprocesoru 8080A

Vstup HOLD přijímá žádost o přenos DMA a vyhovění žádosti je potvrzeno signálem HLDA a uvedením adresové sběrnice do třetího stavu. Signál HLDA pak zajistí i uvolnění datové a řídicí systémové sběrnice, vycházející z obvodu 8228.

Vstup INT přijímá žádost o přerušení. Výstup INTE přenáší informaci o stavu vnitřního klopného obvodu, který povoluje nebo zakazuje přerušení. Stav tohoto klopného obvodu je ovládán signálem RESET a programem.

Výstup WAIT přenáší informaci o zařazení čekacího cyklu TW. Čekací cyklus je zařazen buď na základě žádosti o čekání (signál READY úrovně L), nebo je-li vykonána instrukce HLT – halt.

Zbývající čtyři vývody slouží k napájení. Mikroprocesor 8080A, jako všechny obvody první generace mikroprocesorů, vyžaduje tři napájecí napětí. Používáme-li v systému paměti EPROM 2708 a dynamické paměti DRAM 4116, využijeme všech tří napětí, takže tato skutečnost není na závadu.

Základní zapojení mikroprocesoru 8080A je na obr. 7. Toto zapojení jsem doplnil o obvody, které oddělují a zesilují některé signály tak, aby je bylo možno vyvést na sběrnici. Hodinový obvod 8224, na schématu (obr. 2 a 3) označen jako E₁, je doplněn krystalem X₁ v sérii s kondenzátorem C₂. Sériový kondenzátor se ukázal jako nutný, i když není uváděn v zahraniční literatuře důsledně. Vynecháme-li ho, stává se, že oscilátor nekmitá na jmenovitém kmitočtu krystalu. Kmitočet krystalu jinak není příliš důležitý – nejsme-li omezeni programem, který počítá s určitým kmitočtem (zpoždění pro sériový přenos, akustický výstup, kazetový interface atd.) můžeme použít krystal 12 až 18 MHz. Já mám nejraději 18 MHz, protože pak je cyklus 500 ns a dobře se měří a počítá. V systému JPR-1 se nebude

ze signálu hodin procesoru odvozovat přenosová rychlost sériových portů, takže není nutné respektovat doporučení firmy Intel (18,432 MHz).

Výstup READY 4/E₁ (výstup na vývodu 4 obvodu E₁) je vyveden na špičku 13. Vstup READY 23/E₃ je přiveden ze špičky 12 na špičku 14. Spojení 12–13, které bude ve většině případů, ponechává procesoru E₃ možnost reagovat na signál RDY–3/K1 (špička 3 konektoru sběrnice K1) a zařadit čekací cyklus, není-li READY. Spojení 12–14 způsobí, že procesor nebude reagovat na RDY a zařadí sám jeden čekací cyklus při každém čtení nebo zápisu. Tento režim je vhodný při použití paměti

EPROM s dobou přístupu nad 500 ns.

Vstup RESIN (2/E₁) je přiveden na člen RC, který zajistí nulování systému po zapnutí napájení. Diody D₁ vybíjí rychleji C₁ pro připojení tlačítka RESET – signál RTL (1/K1).

Vstup RDYIN (3/E₁) je upraven odporem R₆ na +5 V a vyveden jako signál sběrnice RDY (3/K1). Výstup RESET (1/E₁) je veden na vstup RESET (12/E₃) procesoru a současně invertován. Vzniklý signál CLR nuluje porty na desce JPR-1 a současně je pod názvem RES vyveden na sběrnici (8/K1). Výstup Φ₂ (TTL) (6/E₁) je vyveden na sběrnici pod názvem Φ₂ (60/K1).

Vstup signálu HOLD – 2/K1 je ošetřen odporem R₃ a invertován obvodem 2/E₂. Výstup invertoru je připojen na vstup HOLD (13/E₃) procesoru. Signál je upraven na správnou úroveň odporem R₄. Výstup HLDA (21/E₃) procesoru je připojen na 2/E₈ a invertován obvodem E₂/10. Vzniklý signál HLDA je vyveden na sběrnici 5/K1 a ještě jednou invertován obvodem 12/E₁₁. Výstupní signál AEN – 13/K1 ovládá zesilovače adresových signálů na sběrnici ARB-1.

Signál D5 (4/E₃) vnitřní datové sběrnice je invertován obvodem 8/E₂ na sběrnici jako M1 – 6/K1. Jako příznak cyklu M1 je tento signál platný pouze po dobu signálu STSTB, který je invertován obvodem 12/E₂ a vyveden na sběrnici (4/K1).

Dekodér adres

V tab. 1 je adresace paměti a portů desky JPR-1. Jako jeden port pracuje i registr povolené úrovně přerušení obvodu 3214. „Kila“ 14., 15. a 16. nejsou na desce využity. V tabulce je pro přehlednost uvedena i adresace paměti desky AND-1.

Jako dekodér adres paměti EPROM pracuje obvod E₁₀. Výstupy E₁₀ ovládají signály CS paměti E₄ až E₇. Vstupy A a B dekodéru volí jednu ze čtyř pamětí, neboli jeden ze čtyř bloků 2K paměťového prostoru procesoru. Výstup dekodéru E₁₀ může však být aktivní pouze za podmínky, že A₁₃ = A₁₄ = A₁₅ = 0 – tato podmínka je splněna pro prvních 8K pamětí. Poslední

Tab. 1. Adresace paměti a portů JPR-1

| „Kilo“ | A15 | A14 | A13 | A12 | A11 | A10 | Začátek | Konec | Vybráno | Obvod |
|--------|-----|-----|-----|-----|-----|-----|---------|-------|-----------|-----------------------------------|
| 1. | 0 | 0 | 0 | 0 | 0 | 0 | 0000 | 03FF | 1. EPROM | E ₇ |
| 2. | 0 | 0 | 0 | 0 | 0 | 1 | 0400 | 07FF | | |
| 3. | 0 | 0 | 0 | 0 | 1 | 0 | 0800 | 0BFF | 2. EPROM | E ₆ |
| 4. | 0 | 0 | 0 | 0 | 1 | 1 | 0C00 | 0FFF | | |
| 5. | 0 | 0 | 0 | 1 | 0 | 0 | 1000 | 13FF | 3. EPROM | E ₅ |
| 6. | 0 | 0 | 0 | 1 | 0 | 1 | 1400 | 1700 | | |
| 7. | 0 | 0 | 0 | 1 | 1 | 0 | 1800 | 1BFF | 4. EPROM | E ₄ |
| 8. | 0 | 0 | 0 | 1 | 1 | 1 | 1C00 | 1FFF | | |
| 9. | 0 | 0 | 1 | 0 | 0 | 0 | 2000 | 23FF | RAM | E ₁₂ , E ₁₃ |
| 10. | 0 | 0 | 1 | 0 | 0 | 1 | 2400 | 27FF | PORT 0 | E ₁₇ , E ₁₈ |
| 11. | 0 | 0 | 1 | 0 | 1 | 0 | 2800 | 2BFF | PORT 1 | E ₁₉ , E ₂₀ |
| 12. | 0 | 0 | 1 | 0 | 1 | 1 | 2C00 | 2FFF | PORT 2 | E ₂₁ , E ₂₂ |
| 13. | 0 | 0 | 1 | 1 | 0 | 0 | 3000 | 33FF | 3214 | E ₁₅ |
| 14. | 0 | 0 | 1 | 1 | 0 | 1 | 3400 | 37FF | | |
| 15. | 0 | 0 | 1 | 1 | 1 | 0 | 3800 | 3BFF | AND-1 | |
| 16. | 0 | 0 | 1 | 1 | 1 | 1 | 3C00 | 3FFF | VIDEO RAM | |

podmínku pro vybrání paměti EPROM „hlídá“ vstup $6/E_{10}$, na němž musí být úroveň H. Na tento vstup je připojen signál MR, invertovaný obvodem $2/E_{11}$. Připojením signálu MR na dekodér adres se předejde konfliktu výstupů paměti EPROM při změně adresy.

Obvod E_9 pracuje jako dekodér adres paměti RAM a portů. Jediná paměť RAM potřebuje všech tisíc adres, pro porty by stačilo po jedné adrese, ale bylo by škoda nevyužít dekodér E_9 , který má 8 výstupů. Za podmínky, že je $A_{15}=A_{14}=0$ a $A_{13}=1$, jsou povoleny výstupy 0 až 7 dekodér E_9 . Tato podmínka je splněna v druhých 8K paměti. Vstupy A, B a C dekodér E_9 určují, které „kilo“ je v tomto prostoru adresováno. Je-li aktivní výstup 0 ($15/E_9$), je vybrána paměť RAM. Hradlo $3/E_{14}$ pracuje jako logický obvod NEBO pro negativní signály MR a MW. Je-li jeden z těchto dvou signálů aktivní, je generován signál MRQ ($3/E_{14}$) = H. Invertorem $8/E_{11}$ a hradlem $11/E_{14}$ je vyroben signál CS pro paměť RAM ($8/E_{12}$ a $8/E_{13}$). Je-li požadováno čtení z paměti RAM, je tento signál postačující. Při zápisu do paměti RAM je nutný ještě signál MW, aktivní při úrovni L a vedený proto přímo z výstupu obvodu E_9 .

Porty, označené pořadovým číslem 0 až 2, jsou dekodovány signály S_0 , S_1 a S_2 ($14/E_9$, $13/E_9$, $12/E_9$). Využívá se zde toho, že obvody 3212 použité jako porty mají jeden povolovací vstup aktivní při H a druhý při L. Signály S_0 až S_2 vybírají vždy dva porty, jeden vstupní a jeden výstupní, a signály MR a MW vyberou správný port podle toho, jedná-li se o vstupní nebo výstupní operaci. Sběrníkové signály MR a MW, vycházející přímo z obvodu 8228, jsou invertovány obvody $2/E_{11}$ a $4/E_{11}$ a označeny jako MR a MW.

Poslední port, jehož adresu je třeba dekodovat, je registr obvodu 3214. Do tohoto registru se pouze zapisuje, a to signálem ESC o úrovni L. Signál je generován hradlem $8/E_{14}$, na jehož jeden vstup je veden MW a na druhý vstup dekodér $11/E_9$, invertovaný hradlem $10/E_{11}$.

Paměti

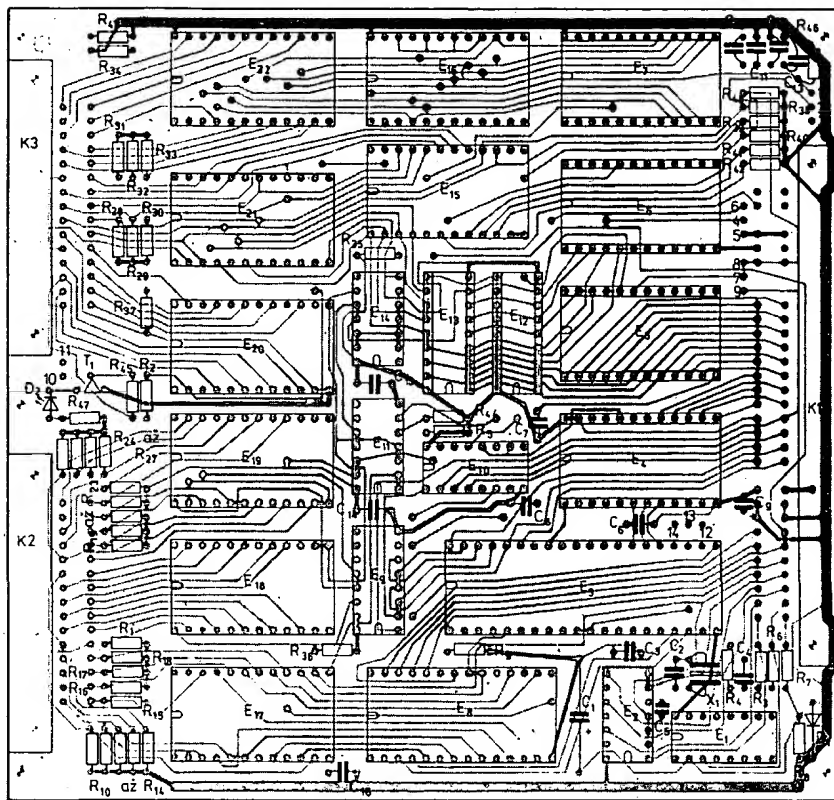
Paměti EPROM se zasouvají do objímek v pozicích E_7 , E_6 , E_5 a E_4 . Umístění objímek na desce je vidět z obr. 8, na němž je rozložení součástek na desce JPR-1. Na obrázku jsou všechny objímky označeny dvojčtou čarou po stranách obvodů. V tab. 1 je v posledním sloupci také adresace obvodů podle polohy na desce. Použijeme-li paměti 2708, musíme spojit špičky 4–5 a 7–8. Použijeme-li paměti 2716, musíme spojit špičky 4–6 a 7–9.

Paměť RAM je tvořena obvody E_{13} a E_{12} . Jsou použity obvody 2114 ($1K \times 4$), které jsou pro malé systémy nejlepší.

Paměťové obvody mají výstupy připojeny přímo na datovou sběrnici mikropočítače JPR-1 a v systému JPR-1 přímo na datovou sběrnici ARB-1. Jsou-li paměti vybrány, pak jejich výstupní zesilovače přímo spínají celou zátěž sběrnice. Zopakujeme si proto povolené zátěže jednotlivých obvodů:

| | |
|------|---------|
| 2708 | 1,6 mA, |
| 2716 | 2,1 mA, |
| 2114 | 2,1 mA. |

Na desce JPR-1 je datová sběrnice zatížena třemi vstupy obvodů 3212 ($3 \times 0,25$ mA) a jedním vstupem obvodu 8228 ($1 \times 0,25$ mA). Proudů ostatních obvodů jsou řádu desítek μA a můžeme je zanedbat. Ve skutečnosti jsou proudy obvodů 3212 a 8228 asi 0,1 mA, protože mají na vstupu tranzistory p-n-p, které mají vždy větší zesílení než uvažované minimální. Jako přijímače dat se na ostat-



Obr. 8. Rozložení součástek na desce JPR-1

ních deskách systému JPR-1 budou používat obvody 3216 nebo 3212, které mají vstupní proudy 0,25 mA. Deska osazená paměťovými obvody může proto spolupracovat v systému se dvěma až čtyřmi deskami dalších pamětí nebo portů. Tam, kde budeme potřebovat desek několik, je nutné paměti vyjmout z desky JPR-1 a jako paměť použít desku REM-1. Program pro větší systém je pak nutno znovu přeložit, protože paměť RAM desky nemůže být na stejném místě jako paměť na desce procesoru. JPR-1 je určen především pro aplikace, při nichž stačí kapacita jeho paměti a portů. Ve dvoudeskové verzi (JPR-1 a AND-1) již pracuje BASIC a tak většině uživatelů postačí základní paměť na desce JPR-1.

Porty

Vstupní i výstupní porty JPR-1 používají obvody 3212. Mnohem rozšířenějšími obvody pro vstup a výstup jsou 8255 nebo 8255A, které jsou programovatelné. Tyto obvody však potřebují oddělovací zesilovače na vstupech i výstupech a s nimi je nutná plocha desky s plošnými spoji pro stejný počet vstupů a výstupů větší než při obvodech 3212, které jsou bipolární a mají výkonovější výstupy a vstupy odolnější proti zničení. Ideálním obvodem pro malé systémy je TMS5501, vyráběný firmou TI. Tento obvod nahrazuje funkci systémového řadiče 8228 a má následující parametry: dvě linky pro externí přerušení, 8bitový vstupní port, 8bitový výstupní port, jeden sériový port (UART) a pět programovatelných časovačů. Obvod TMS5501 je v pouzdře se 40 vývody a použije-li se s obvody 8224 a 8080A, pak téměř nahradí systém jednočipových mikročipů.

Obvod 3212 byl i v naší literatuře dostatečně popsán, takže není nutné se jím blíže zabývat. Deska JPR-1 má 3 vstupní porty a 3 výstupní porty. Každý z portů je osmibitový. Konektory pro vstup a výstup z desky mají pouze po 30 špičkách a tak

nebylo možné vyvést všechny porty standardně – jejich zapojení se proto nepatrně liší.

Vstupní port P_0 (E_{17}) má kromě osmi vstupů vyveden na konektor K_2 i vstup STB ($11/E_{17}$). Obvod 3212 není registr, ale „latch“ (těžko přeložitelné slovo), který pracuje tak, že je-li STB na úrovni H, informace klopnými obvody pouze prochází. Jakmile se změní STB na L, zůstane v klopných obvodech uchována poslední informace, která byla na jeho vstupech. (Srovnej s obvodem 7475 = latch a 7474 = registr). Vstup STB portu 0 je ošetřen odporem R_{10} , takže je-li nezapojen, má STB úroveň H. Připojujeme-li k JPR-1 zařízení, které má informaci platnou jen po dobu danou signálem, jež potvrzuje platnost (vodící stopa snímáče pásy, DAV, SC atd.), můžeme takový signál použít jako STB pro port 0. Tento port, stejně jako všechny ostatní, je nulován signálem CLR, což je vlastně RESET mikroprocesoru. Tento signál ovšem vynuluje klopné obvody portu P_0 jen tehdy, je-li STB úrovně L.

Vstupy všech vstupních portů jsou ošetřeny odpory, aby byla definována úroveň na nezapojených vstupech.

Vstupní port P_1 je zapojen prakticky stejně. Navíc je vyveden na konektor K_2 výstup INT ($23/E_{16}$) obvodu 3212. Výstup INT obvodu 3212 je aktivní při L. Výstup INT se nastaví po skončení signálu STB, neboli oznamuje, že klopné obvody mají novou informaci. Právě proto se nazývá INT, protože se používá k přerušení po zapsání nové informace. Při používání portu P_1 však nás tento význam signálu INT nebude zajímat. Signál STB dáme my, z okolí mikročipů a víme tedy, kdy skončil, a kdy je informace nová. My využijeme druhé vlastnosti výstupu INT. Je-li tento signál nastaven (je-li jeho úroveň L), čeká se, až počítač převezme novou informaci u JPR-1 přečtením

paměti na adrese portu P1. Jakmile skončí signál MR, který čtení z paměti doprovází, skončí INT portu, čímž vlastně procesor „poděkuje“ za novou informaci. Signál INT tedy můžeme použít k synchronizaci předávání dat do portu P1. Zapišme data, úroveň INT se změní na L, počítač data převezme, úroveň INT se změní na H a vše se opakuje.

Na vstupní port (E₂₁) nezbyly již žádné špičky pro STB a INT. Proto P2 vlastně pouze odděluje data od sběrnice a je ovládán adresou portu (u JPR-1 je to adresa paměti) a signálem MR. Protože nezbyly špičky ani vstupy, které vyvolávají přerušení, je port P2 použit jako vstupní port a současně jako vstup osmi přerušovacích vstupů. Pozor na to, že vstupy P2-IN 0 a P2-IN 1 jsou současně vyvedeny i na sběrnici. K těmto dvěma vstupům se musíme připojit pouze hradly s otevřeným kolektorem (7403, 7405, kolektor tranzistoru nebo kontakt), neboť mohou být na sběrnici využity jinými deskami systému (AND-1 používá INT 1). Dvě funkce tohoto portu nejsou však na závadu – nepoužíváme-li přerušení vůbec, dostaneme běžný vstupní port. Používáme-li přerušení, můžeme se programově přesvědčit, zda ještě trvá žádost o přerušení, než přerušení povolíme znovu. Navíc si můžeme napsat testovací program, který díky dvojí funkci vstupů otestuje správnou funkci přerušovacího systému JPR-1.

Vstupní porty JPR-1 se také trochu odlišují, opět ze stejných důvodů (nedostatek špiček konektorů). Výstupní port P0 má kromě datových výstupů vyveden ještě signál INT. U vstupních portů jsme se seznámili s dvěma funkcemi tohoto výstupu. Je snad ještě třetí funkce? Ano, je: výstup INT je vyveden jako výstup vnitřního hradla NEBO a jakýkoli signál o úrovni H na jeho vstupu způsobí vznik signálu INT. Na jeden vstup tohoto hradla NEBO vede vnitřní klopný obvod INT, který „dělá“ ty dvě, již známé funkce. Na druhý vstup vede vnitřní signál, aktivní po dobu, kdy je obvod 3212 vybrán (u JPR-1 po dobu signálu MW, neboť se jedná o výstupní port). U výstupního portu P0 se STB vůbec nepoužívá a tak se vnitřní klopný obvod INT vůbec neuplatní. Na výstupu INT však vznikne impuls délky asi 500 ns po dobu, po níž probíhá zápis do tohoto portu. Dnes již nikdo nezjistí, zda to návrhář obvodu 3212 chtěl. Původním posláním tohoto vnitřního spoje v obvodu (spoj SEL na NOR) je blokovat skončení INT, než skončí výběr obvodu, neboť klopný obvod se nuluje hned při vybrání (DS1 = 0 a DS2 = 1). My však můžeme tohoto impulsu využít jako informace, že procesor posílá nová data. Pozor však na délku tohoto impulsu, která je příliš krátká na dlouhé vodiče a rušení nejde u tak krátkých impulsů filtrovat!

Výstupní port P1 má vyvedeno jak STB, tak INT, a proto se tedy uplatní všechny tři funkce signálu INT. Stejně je zapojen i výstupní port P2. Všechny výstupní porty pracují ve „výstupním módu“ (MD = 1) a mají tedy trvale otevřeny výstupní trístavové budíče. Výstupy obvodu 3212 jsou schopny spínat při úrovni L zátěž až 15 mA a při H (3,65 V) dodávat proud 1 mA.

Obsah klopných obvodů všech výstupních portů je nulován opět signálem CLR.

Přerušení

Obvod 3214 (E₁₅), 3212 (E₁₆), instrukce pro povolení a zakázání přerušení (EI a DI), vnitřní klopný obvod IE procesoru 8080A, výstup INTA obvodu 8228 a vstupy portu P2 tvoří základ tzv. systému přeru-

Tab. 2. Význam datových bitů zapisovaných do registru obvodu 3214

Registr povolené úrovně přerušení

adresa: 3000H
adresace: MW

| D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 | |
|----|----|----|----|-----|----|----|----|--|
| | | | | SGS | B2 | B1 | B0 | |
| | | | | 0 | 0 | 0 | 0 | zakázáno jakékoli přerušení |
| | | | | 0 | 0 | 0 | 1 | povoleno: R7 |
| | | | | 0 | 0 | 1 | 0 | povoleno: R7, R6 |
| | | | | 0 | 0 | 1 | 1 | povoleno: R7, R6, R5 |
| | | | | 0 | 1 | 0 | 0 | povoleno: R7, R6, R5, R4 |
| | | | | 0 | 1 | 0 | 1 | povoleno: R7, R6, R5, R4, R3 |
| | | | | 0 | 1 | 1 | 0 | povoleno: R7, R6, R5, R4, R3, R2 |
| | | | | 0 | 1 | 1 | 1 | povoleno: R7, R6, R5, R4, R3, R2, R1 |
| | | | | 1 | 0 | 0 | 0 | povoleno: R7, R6, R5, R4, R3, R2, R1, R0 |
| | | | | 1 | X | X | X | povoleno: R7, R6, R5, R4, R3, R2, R1, R0 |

šení mikropočítače JPR-1. Přerušovací systém je zapojen přesně podle katalogů a příruček firmy INTEL (podle těch prvních, dnes již se s obvodem 3214 v literatuře nesetkáme; nový obvod 8259 vytlačil svého předchůdce).

Přerušovací systém je značně závislý na programech a proto si popíšeme jen to nejnужnější. Procesor 8080A má pouze jeden vstup přerušení, lépe řečeno vstup žádosti o přerušení! To je na mikropočítač poměrně málo a tak musí pomoci obvod 3214.

Obvod 3214 má 8 vstupů žádostí o přerušení. Tyto vstupy jsou aktivní při úrovni L. Procesor však může vyhovět pouze jedné žádosti. Aby se jednotliví žadatelé o přerušení mezi sebou nepoprali, je v obvodu dekodér, který určí podle žádosti na vstupech R0 až R7, kdo bude mít přednost. Nejvyšší prioritu má vstup R7. Vstup R6 je hned za ním a tak dále. Vstup R0 je ten „poslední vřadu“. Prioritní dekodér vydává tříbitový kód na vstupech A0, A1 a A2 obvodu 3214. Kód A2, A1, A0 je číslo, označující nejvyšší úroveň z těch, které žádají současně o přerušení. Protože procesor 8080A má 8 instrukcí RESTART (RST *n*) se strojovým kódem 11NNN111, stačí do bitů, v nichž je psáno *N*, přičíst výstupy A2, A1, A0 obvodu 3214 a na ostatní dát log. 1 (H). Získáme tak 8 strojových kódů pro 8 instrukcí, které pak můžeme ve vhodný okamžik nabídnout procesoru 8080A ke zpracování.

Tříbitový výstup prioritního dekodéru však neslouží pouze pro určení čísla instrukce RST *n*. Obvod 3214 má ještě uvnitř tzv. komparátor priority. Tento komparátor porovnává úroveň žádosti o přerušení s obsahem registru povolení úrovně přerušení. Tento registr je také uvnitř obvodu a u JPR-1 lze jeho obsah změnit zápisem na adresu 3000H. Zapisují se pouze bity D3, D2, D1 a D0 a význam těchto bitů je v tab. 2. Při studiu materiálu o tomto obvodu vznikají nejasnosti, protože vstupy obvodu 3214 jsou invertující. Zapamatujte si proto pravidlo, že do registru povolených přerušení obvodu 3214 se posílá: *N* = počet přerušení (vstupů), která mají být povolena. Pošleme-li nulu, nebude povoleno žádné přerušení, pošleme-li pětku, bude jich povoleno pět atd. Pak stačí vědět, že sedmička má nejvyšší úroveň a vždy si odvodíme tab. 2.

Přerušovací systém s obvodem 3214 má kromě registru povolených přerušení ještě dvě možnosti, jak povolit, nebo zakázat libovolné množství přerušení. Do obvodu 3214 vede výstup INTE z procesoru 8080A (spoj 16/E₃ na 7/E₁₅). Není-li přerušení povoleno programem (instruk-

ce EI), nereaguje obvod 3214 na žádnou žádost o přerušení.

V případě, že jsou splněny všechny podmínky pro přerušení (žádost na vstupu R0 až R7, odpovídající obsazení registru povolených přerušení a INTE z procesoru je log. H), vydá obvod 3214 na výstupu INT (5/E₁₅) krátký negativní impuls, který je invertován obvodem 6/E₁₁ a přiveden na STB obvodu 3212 (11/E₁₆). Obvod 3212 slouží v přerušovacím systému jako paměť pro kód vydávaný výstupy A2, A1 a A0 obvodu 3214 a také jako paměť, která si zapamatuje, že přišel impuls INT (trvá jen jednu periodu Φ₂). Vydáním kódu a signálu INT se však automaticky obvod 3214 zablokuje a nereaguje na další žádosti o přerušení až do doby, kdy se realizuje opětovný zápis do registru povolené úrovně přerušení (zápis na adresu 3000H). Tím jsme si objasnili způsob povolování a zakazování přerušení. Podívejme se nyní na to, co se děje se žádostí o přerušení dále. Tím, že jsme zapsali signál pomocí vstupu STB (11/E₁₆) do obvodu 3212, zapsali jsme kromě kódu i další informace na vstupech D7, D6 a D1, D2, D3. Na tyto vstupy je přivedena úroveň H (odpor R₃₅ na +5 V) a obsah dočasné paměti kódu žádosti o přerušení bude:

1 1 A2 A1 A0 1 1 1

Po skončení signálu STB se nastaví výstup INT obvodu 3212 na L (23/E₁₆), je invertován hradlem 6/E₁₁, vznikne signál INT a ten již jde na procesorový vstup INT (14/E₃). Odpor R₅ upravuje výstup obvodu TTL na správnou úroveň H. Žádost o přerušení zpracovává procesor 8080A buď hned (je-li ve stavu HALT po instrukci HLT), nebo po dokončení právě prováděné instrukce. Samozřejmě předpokládáme, že je přerušení povoleno, jinak by vlastně ani obvodem 3214 žádost neprošla.

Procesor 8080A zpracovává přerušení velice chytře. Nejprve pošle na vnitřní datovou sběrnici stavový byte, podobný stavovému byte cyklu čtení instrukce z paměti (FETCH). Stavový byte obsahuje příznaky M1 a INTE (u FETCH místo INTA je MR) a v obvodu 8228 vznikne signál INTA (23/E₃), který je časově shodný se signálem MR při cyklu FETCH. Dá se říci, že obslužit přerušení znamená vlastně číst instrukce doprovázené speciálním signálem INTA (místo obvyklého MR). Rozdíl je v tom, že před vydáním INTA „shodí“ procesor signál INTE a zablokuje tak přerušovací systém. Výstup INTA obvodu 8228 je aktivní při L a je veden na vstup DS1 (1/E₁₆). Tím se obsah dočasné paměti kódu žádosti o přerušení dostane

Tab. 3. Zpracování žádosti o přerušení u systému JPR-1

| Žádost = log. 0 (L) | | | | | | Obsah 3212, E ₁₆ | | | | | | | | | |
|---------------------|----|----------|----|------------|-----------------|-----------------------------|----|----|----|----|----|----|----|-----------|--------|
| Vstup | K3 | Sběrnice | K1 | Vstup 3214 | E ₁₅ | D7 | D6 | D5 | D4 | D3 | D2 | D1 | D0 | Instrukce | Adresa |
| P2 - IN0 | 20 | INT 0 | 62 | R0 | 15 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | RST 7 | 0038H |
| P2 - IN1 | 17 | INT 1 | 59 | R1 | 16 | 1 | 1 | 1 | 1 | 0 | 1 | 1 | 1 | RST 6 | 0030H |
| P2 - IN2 | 12 | - | - | R2 | 17 | 1 | 1 | 1 | 0 | 1 | 1 | 1 | 1 | RST 5 | 0028H |
| P2 - IN3 | 14 | - | - | R3 | 18 | 1 | 1 | 1 | 0 | 0 | 1 | 1 | 1 | RST 4 | 0020H |
| P2 - IN4 | 13 | - | - | R4 | 19 | 1 | 1 | 0 | 1 | 1 | 1 | 1 | 1 | RST 3 | 0018H |
| P2 - IN5 | 18 | - | - | R5 | 20 | 1 | 1 | 0 | 1 | 0 | 1 | 1 | 1 | RST 2 | 0010H |
| P2 - IN6 | 15 | - | - | R6 | 21 | 1 | 1 | 0 | 0 | 1 | 1 | 1 | 1 | RST 1 | 0008H |
| P2 - IN7 | 16 | - | - | R7 | 22 | 1 | 1 | 0 | 0 | 0 | 1 | 1 | 1 | RST 0 | 0000H |

A2 A1 A0 ← kód 3214

na datovou sběrnici a procesor ho přijme jako instrukci. Protože kód je sestaven tak, aby tvořil instrukce RST n , provede procesor tuto instrukci a program tak může pokračovat ve vykonávání programu na jedné z osmi možných adres, které jsou určeny číslem instrukce RST. V tab. 3 je znázorněno, jak žádost na některém z osmi přerušovacích vstupů JPR-1 (P2 - IN0 až P2 - IN7) nebo na dvou přerušovacích vstupech sběrnice ARB-1 (INT 0 a INT 1) vytvoří instrukci RST n a na jaké adrese bude program pokračovat. Kdo by si chtěl zapamatovat vztah žádosti k číslům instrukcí RST, ať si zapamatuje, že žádost 0 vyvolá RST 7 a žádost 7 vyvolá RST 0.

Systém 8080A má i další možnosti jak zpracovat přerušení. Rozpojíme-li spojku 1-2 na desce JPR-1 a spojíme 1-3, dostane se na výstup INTA (23/E₈) napětí +12 V přes odpor 1 kΩ (R₄₈). Obvod 8228 to pozná a vyšle na vnitřní datovou sběrnici instrukci RST7 jako odpověď na cyklus zpracování přerušení INTA. Další možnost systému 8080A spočívá ve zpracování instrukce CALL (3 byte). Tento trik se používá ve spojení s řadičem přerušení 8259. Kdo by si chtěl s tímto obvodem něco vyzkoušet v systému JPR-1, nesmí zapomenout, že se signál INT na procesor 8080A dostává přes hradlo 6/E₁₄ a musí použít buď oba, nebo jeden jeho vstup.

Na závěr kapitoly o přerušení si ještě uvedme a zopakujeme několik pravidel:

1. Žádost o přerušení u obvodu 3214 musí být aktivní (tj. úrovně L), dokud není akceptována. Obvod nemá vstupy ovládané hranou signálu! Někdy je to problém, neboť okolí mikropočítače se ani nedozví o tom, že byla žádost zpracována a potvrzena signálem INTA. Proto je někdy nutné použít výstupní port a programově zajistit potvrzení, že žádost již byla zpracována.
2. Opačný problém může vzniknout, je-li žádost aktivní i po jejím zpracování. Může se pak stát, že tentýž požadavek způsobí několikanásobné přerušení.
3. Po každém zpracování žádosti o přerušení nesmíme zapomenout na znovupovolení přerušení instrukcí EI (systém sám provedl DI při INTA) a nahrání dat do registru obvodu 3214 (adresa 3000H).
4. Po zapnutí systému musí program realizovat počáteční nastavení obsahu registru obvodu 3214, neboť tento obvod nemá vstup RESET.
5. Nejlépe se přerušovací systém JPR-1 zkouší na přípravku TST-3 s diodovou pamětí EPROM TST-02. Tato paměť má pouze 8 byte a čte se stále dokola (stejný obsah na každých 8 adresách po sobě). Napišeme-li krátký program (FB, 3E, 04, 32, 00, 30, C7), můžeme zkoušet přerušení, aniž bychom měli na adresách, na něž skáče RST, obsluhové programy, neboť RST vrátí program zpět na první instrukci v paměti TST-02.

Oživení desky procesoru JPR-1

Při ožívání desky (obr. 8, 9 a 10) oceníme skutečnost, že JPR-1 nemá oddělenou sběrnici zesilovači adresových, datových a hlavních řídicích signálů. Jakou to má pro ožívání výhodu? Prakticky u všech jednodeskových mikropočítačů jsou například na adresách jednosměrně zesilovače adresových signálů. Při ožívání takto navržených desek nelze vnutit ze sběrnice adresu pamětem nebo portům. U JPR-1 jsou signály sběrnice na konektoru K1 vlastně vstupy a výstupy současně. Neosadíme-li na desce procesor (E₃), hodinový obvod 8224 (E₁) a systémový řadič 8228 (E₂), vznikne z JPR-1 deska paměti a portů a všechny adresové, datové a řídicí signály nutné pro její činnost jsou na konektoru K1 vlastně v opačné funkci, než u procesoru. Tam, kde byl výstup adresy, je nyní vstup a tam, odkud vycházel signál MR, tam bude jeho vstup.

Při ožívání desky začneme tedy tím, že neosadíme tři jmenované obvody, a na-

víc necháme prázdné i objímky paměti E₄, E₅, E₆, E₇, E₁₂ a E₁₃. Desku zasuneme do konektoru K1 přípravku TST-03 a přepínačem ENABLE na tomto přípravku povolíme výstup zesilovačů adresových a řídicích signálů. Pomocí adresových přepínačů přezkoušíme, vedou-li adresové signály správně na paměti i na dekodéry E₉ a E₁₀. Navíc můžeme přezkoušet, zda tyto signály vedou správně i na objímku procesoru E₃.

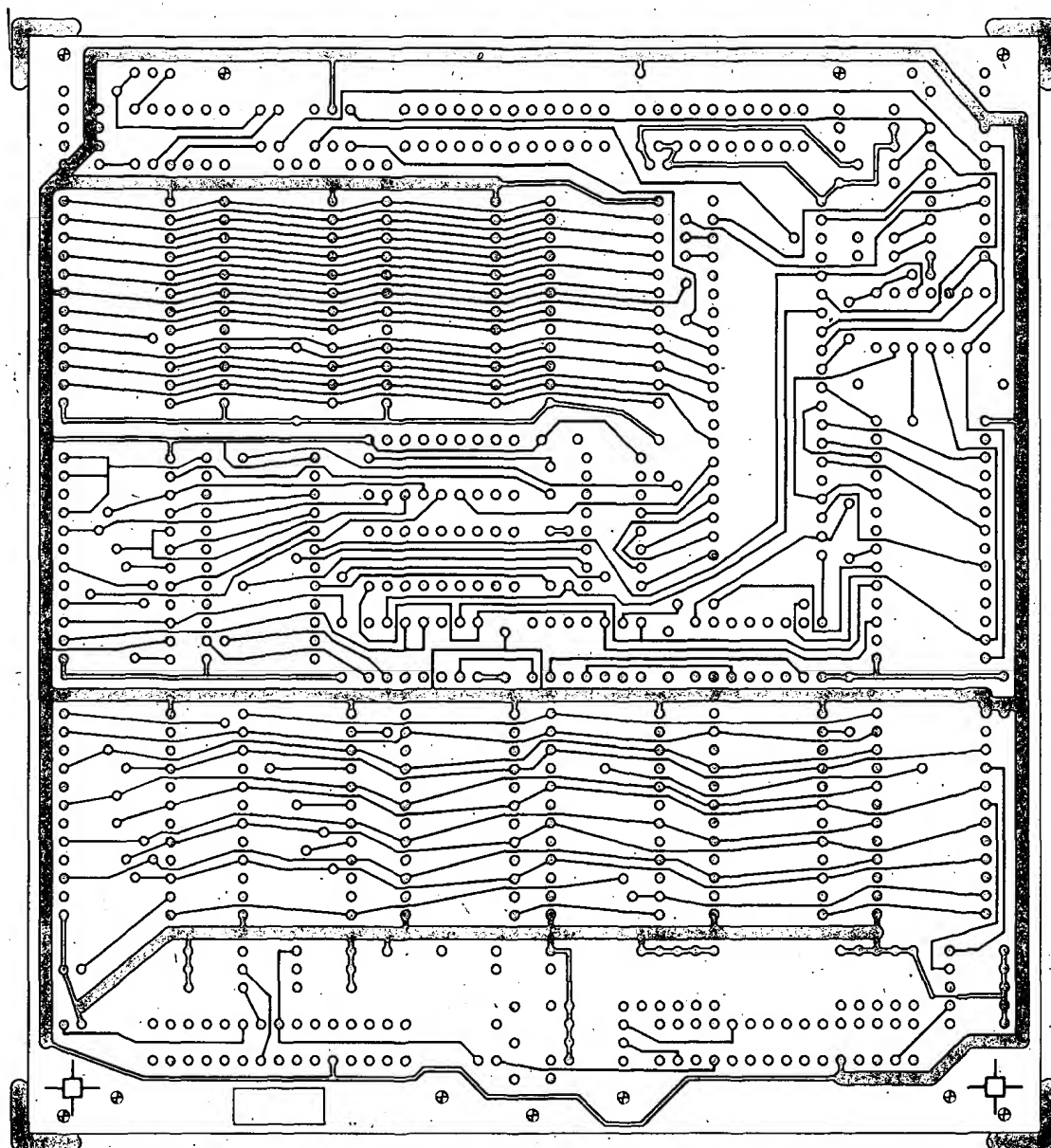
Potom si vezmeme k ruce tab. 4 s adresací jednotlivých pamětí a portů na desce JPR-1. Nastavíme na přípravku adresu a pak stlačíme tlačítko MR nebo MW. V tab. 4 jsou uvedeny jednotlivé měřicí body (vývod obvodu lomeno pozicí obvodu na desce) a údaje, které musíme naměřit logickou sondou. Nesouhlasí-li něco, není obvykle problém závadu najít, protože mezi přípravkem a měřeným bodem je jen konektor a jeden až dva logické obvody. Tímto postupem ověříme dekodéry adres.

Pracuji-li dekodéry adres správně, můžeme začít ožívat paměti a porty. Paměti EPROM nahradíme přípravkem TST-02 (diodová paměť EPROM 8 × 8 bit) a pomocí tlačítka MR na přípravku TST-03 čteme data. Na TST-02 nastavíme různé kombinace dat a ověříme, zda se čtou samé L, samé H a konstanty AA a 55. Tím prověříme datové cesty přes konektor K1 do přípravku. Paměti RAM zasuneme do objímek a zkusíme do nich zapsat a přečíst jejich obsah. Všechny buňky samozřejmě nemůžeme otestovat a tak necháme další testování až na program.

Porty se nejlépe zkoušejí simulátorem vstupů a výstupů (V/V). Je to přípravek, který má tolik přepínačů, kolik má JPR-1 vstupů (3 × 8 a 4 × STB), a tolik indikátorů tvořených invertorem a diodou LED, kolik má JPR-1 výstupů (3 × 8 a 4 × INT). Přípravek připojíme kabely na konektory K2 (tab. 5) a K3 (tab. 6). Přípravkem TST-03 a simulátorem V/V pak proměříme funkci všech vstupů a výstupů JPR-1. Přerušovací vstupy prověříme například progra-

Tab. 4. Měřicí body pro ožívání adresace JPR-1

| Adresa | Vybráno | Tlačítko | „Nula“ (L) | „Jednička“ (H) |
|--------|---------------------------------------|----------|---|--|
| 0000 | EPROM E ₇ | MR | 15/E ₁₀ ; 20/E ₇ | |
| 07FF | EPROM E ₇ | MR | 15/E ₁₀ ; 20/E ₇ | |
| 0800 | EPROM E ₆ | MR | 14/E ₁₀ ; 20/E ₆ | |
| 0FFF | EPROM E ₆ | MR | 14/E ₁₀ ; 20/E ₆ | |
| 1000 | EPROM E ₅ | MR | 13/E ₁₀ ; 20/E ₅ | |
| 1700 | EPROM E ₅ | MR | 13/E ₁₀ ; 20/E ₅ | |
| 1800 | EPROM E ₄ | MR | 12/E ₁₀ ; 20/E ₄ | |
| 1FFF | EPROM E ₄ | MR | 12/E ₁₀ ; 20/E ₄ | |
| 2000 | RAM E ₁₂ , E ₁₃ | MW | 15/E ₉ ; 1/E ₁₄ ; 11/E ₁₄ ; 8 a 10/E ₁₂ a E ₁₃ | 8/E ₁₁ ; 3/E ₁₄ ; 12/E ₁₄ |
| 23FF | RAM E ₁₂ , E ₁₃ | MW | 15/E ₉ ; 1/E ₁₄ ; 11/E ₁₄ ; 8 a 10/E ₁₂ a E ₁₃ | 8/E ₁₁ ; 3/E ₁₄ ; 12/E ₁₄ |
| 2000 | RAM E ₁₂ , E ₁₃ | MR | 15/E ₉ ; 2/E ₁₄ ; 11/E ₁₄ ; 8/E ₁₂ a E ₁₃ | 8/E ₁₁ ; 3/E ₁₄ ; 12/E ₁₄ |
| 23FF | RAM E ₁₂ , E ₁₃ | MR | 15/E ₉ ; 2/E ₁₄ ; 11/E ₁₄ ; 8/E ₁₂ a E ₁₃ | 8/E ₁₁ ; 3/E ₁₄ ; 12/E ₁₄ |
| 2400 | PORT 0 E ₁₇ | MR | 14/E ₉ ; 1/E ₁₇ ; 1/E ₁₁ | 2/E ₁₁ ; 13/E ₁₇ |
| 2400 | PORT 0 E ₁₈ | MW | 14/E ₉ ; 1/E ₁₈ ; 3/E ₁₁ | 4/E ₁₁ ; 13/E ₁₈ |
| 2800 | PORT 1 E ₁₉ | MR | 13/E ₉ ; 1/E ₁₉ ; 1/E ₁₁ | 2/E ₁₁ ; 13/E ₁₉ |
| 2800 | PORT 1 E ₂₀ | MW | 13/E ₉ ; 1/E ₂₀ ; 3/E ₁₁ | 4/E ₁₁ ; 13/E ₂₀ |
| 2C00 | PORT 2 E ₂₁ | MR | 12/E ₉ ; 1/E ₂₁ ; 1/E ₁₁ | 2/E ₁₁ ; 13/E ₂₁ |
| 2C00 | PORT 2 E ₂₂ | MW | 12/E ₉ ; 1/E ₂₂ ; 3/E ₁₁ | 4/E ₁₁ ; 13/E ₂₂ |
| 3000 | 3214 E ₁₅ | MW | 11/E ₉ ; 3/E ₁₁ ; 8/E ₁₄ ; 23/E ₁₅ | 10/E ₁₁ ; 9/E ₁₄ |



Obr. 9. Deska s plošnými spoji JPR-1, strana součástek

mem, který jsem uvedl při popisu přerušovacího systému.

Další postup ožívování závisí na tom, je-li k dispozici simulátor paměti EPROM. Simulátorem můžeme vyzkoušet krátké i delší programy, aniž bychom programovali a mazali paměti EPROM. Nemáme-li simulátor, musíme vyzkoušet JPR-1 třeba s programem Monitor nebo BASIC.

Ožívujeme-li další desky systému, můžeme využít přípravku TST-03 a kabelu KB-02 ve funkci tlačítkového mikroprocesoru. Kabel zapojíme jedním koncem do TST-03 a druhým do objímek obvodů 8080A a 8228; pak můžeme vyzkoušet nejen paměti a porty na desce JPR-1, ale i na ostatních deskách systému přes sběrnici ARB-1.

Přípravek TST-03 je vhodný i k ožívování krátkých programů nebo k hledání chyb; o nichž nevíme, jsou-li v HW nebo SW: Při zkoušení samotných desek je sepnut přepínač EN, který povoluje vysílání adresových, datových a řídicích signálů. Rozpojíme-li tento přepínač, jsou vysílání ovládány signálem HLDA (5/K1 sběrnice). Potom můžeme do přípravku zasunout desku procesoru plně osazenou

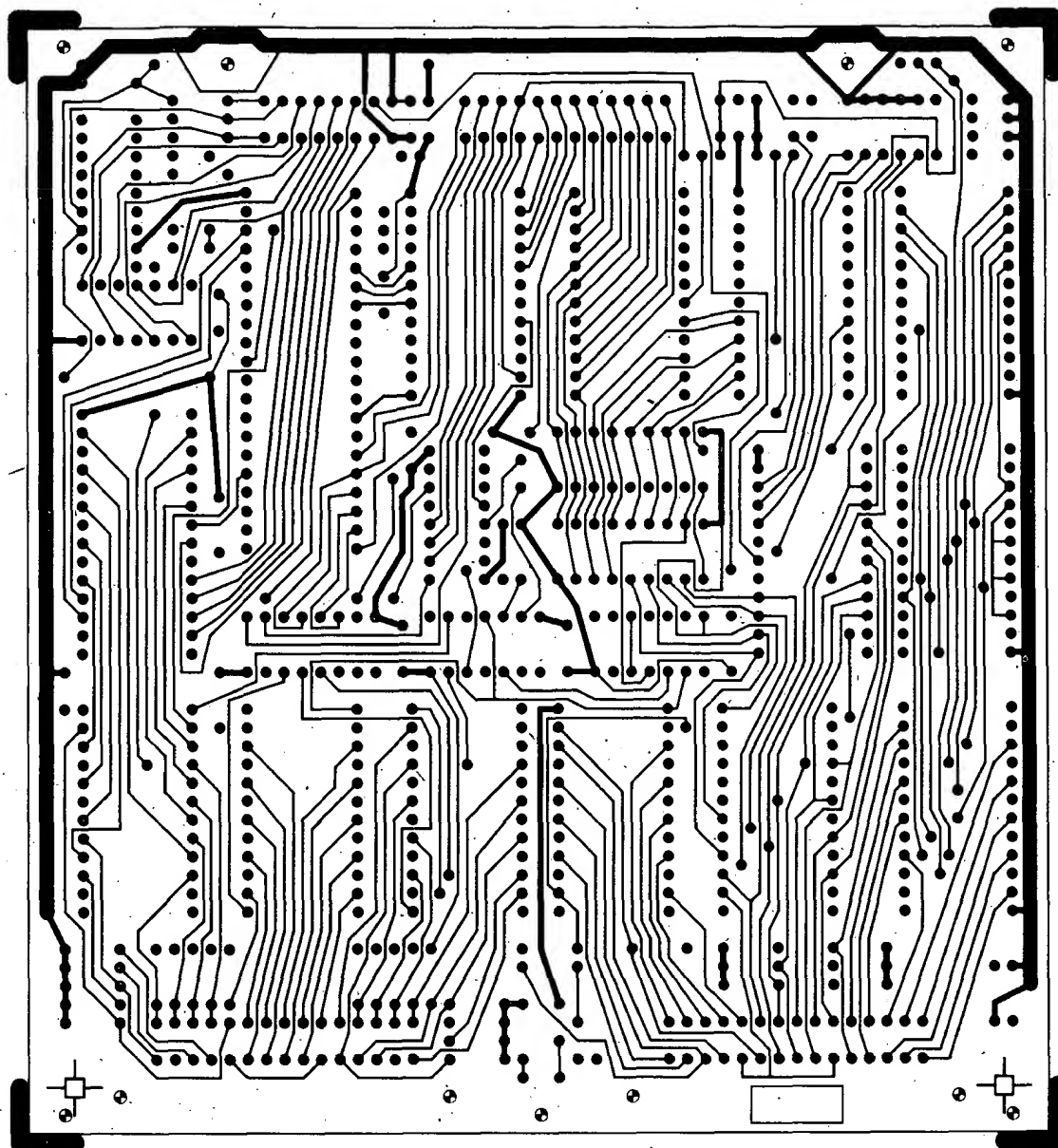
všemi obvody. Chceme-li něco zapsat do portu nebo paměti, požádáme procesor o DMA přepínačem HOLD. Jakmile se rozsvítí LED DMA, můžeme ovládat řídicí signály adresy i data my, neboť procesor nám zapůjčil sběrnici. Dále je možné krokovat chod procesoru (přepneme přepínač STEP/RUN na STEP) a to po instrukčních cyklech (přepínač MC/IC na IC) nebo po strojových cyklech (MC).

Přípravek indikuje všechny důležité signály, takže dobře informuje o tom, co procesor právě dělá. Kabelem KB-03 pak můžeme propojit přípravek TST-03 se sběrnici ARB-1 a ožívovat tak celý systém JPR-1 najednou.

Pomocí jednoduchých přípravků není samozřejmě možné najít všechny chyby, které se mohou při ožívování nebo poruše systému vyskytnout. Proto je třeba znát

Tab. 5. Zapojení konektoru K2 desky JPR-1

| Č. | Signál | Název | Typ. | Č. | Signál | Název | Typ |
|------------------|---------|------------------|-----------|-------------|-------------|------------------|--------------|
| 1 | P1-IN1 | vstupní port P1 | OUT | 2 | P1-IN4 | vstupní port P1 | INP |
| 3 | P1-IN2 | | INP | 4 | P1-IN1 | | INP |
| 5 | P1-IN3 | | INP | 6 | P1-IN0 | | INP |
| 7 | P1-IN7 | | INP | 8 | P1-IN5 | | INP |
| 9 | P1-IN6 | | INP | 10 | P1-STB | | INP |
| 11 | 0 V | zem | | 12 | P0-IN7 | | OUT |
| 13 | P0-OUT0 | výstupní port P0 | OUT | 14 | P0-OUT1 | výstupní port P0 | OUT |
| 15 | P0-OUT5 | | OUT | 16 | P0-OUT6 | | OUT |
| 17 | P0-OUT4 | | OUT | 18 | P0-OUT7 | | OUT |
| 19 | P0-OUT3 | | OUT | 20 | P0-OUT2 | | OUT |
| 21 | 0 V | | zem | | 22 | | P0-STB |
| 23 | P0-IN4 | vstupní port P0 | INP | 24 | P0-IN6 | vstupní port P0 | INP |
| 25 | P0-IN7 | | INP | 26 | P0-IN5 | | INP |
| 27 | P0-IN2 | | INP | 28 | P0-IN1 | | INP |
| 29 | P0-IN3 | | INP | 30 | P0-IN0 | | INP |
| Číslo konektoru: | | | K2 | Konektor: | TY 513 3011 | | OUT – výstup |
| Deska/zařízení: | | JPR-1 | Protikus: | TX 514 3013 | | INP – vstup | |
| Klíčování: | | F-3 | | | | | |



Obr. 10. Deska s plošnými spoji JPR-1, spodní strana

průběhy jednotlivých signálů. Některé základní časové průběhy již byly uvedeny v kapitole o sběrnici ARB-1. Zde bych chtěl spíše ukázat, jak se k měření časových signálů v mikroprocesorových systémech přistupuje. Při měření osciloskopem nebo logickým analyzátozem je výhodné, je-li průběh periodický, neboť se lépe synchronizuje a lépe se vyznáme v tom, co na obrazovce sledujeme. U počítačů však není jednoduché „vyrobit“ pe-

riodický průběh. Je nutné najít co nejkratší „program“, ten pak „zacyklit“ a najít vhodnou synchronizaci.

Obr. 11 a 12 ukazují dva základní cykly procesoru 8080A. Na obr. 11 je zápis do paměti a na obr. 12 čtení z paměti, ale s jedním čekacím cyklem T_w (spojení WAIT-READY). Proto je na každém obrázku znázorněno, při jakém programu se měřilo. Časové průběhy byly měřeny logickým analyzátozem Schlumberger

7600. Na každém obrázku jsou znázorněny i body, v nichž byly průběhy měřeny.

Seznam součástek

Odporů (TR 191 $\pm 10\%$, označení K)

R1, R2, R4,
R5, R8, R34,
R36, R45
R3, R6,
R35,
R40 až R42,
R44, R48
R7, R9
R10 až R33,
R37 až R39,
R43, R46
R47

10 k Ω

1 k Ω

100 Ω

4,7 k Ω

220 Ω

Kondenzátory

C1 20 μ F/6 V, TE 981
C2 10 pF, TK 755
C3 až C8,
C10 až C16 22 nF, TK 783
C9 4,7 μ F/6,3 V, TE 121

Integrované obvody

E1 MH8224
E2, E11 MH7404

Tab. 6. Zapojení konektoru K3 desky JPR-1

| Č. | Signál | Název | Typ | Č. | Signál | Název | Typ |
|------------------|---------|--------------------------------|-------|-----------|---------|--------------------------------|--------------|
| 1 | P2-INT | výstupní port P2 | OUT | 2 | P2-OUT5 | výstupní port P2 | OUT |
| 3 | P2-OUT0 | | OUT | 4 | P2-OUT6 | | OUT |
| 5 | P2-OUT1 | | OUT | 6 | P2-OUT2 | | OUT |
| 7 | P2-OUT7 | | OUT | 8 | P2-OUT3 | | OUT |
| 9 | P2-STB | | INP | 10 | P2-OUT4 | | |
| 11 | 0 V | zem | | 12 | P2-IN2 | | INP |
| 13 | P2-IN4 | vstupní port P2 + přerušení | INP | 14 | P2-IN3 | vstupní port P2 + přerušení | INP |
| 15 | P2-IN6 | | INP | 16 | P2-IN7 | | INP |
| 17 | P2-IN1 | | INP | 18 | P2-IN5 | | INP |
| 19 | P1-STBO | | INP | 20 | P2-IN0 | | INP |
| 21 | 0 V | | zem | | 22 | | P1-INTR |
| 23 | P1-OUT4 | výstupní port P1 | OUT | 24 | P1-OUT0 | výstupní port P1 | OUT |
| 25 | P1-OUT7 | | OUT | 26 | P1-OUT1 | | OUT |
| 27 | P1-OUT2 | | OUT | 28 | P1-OUT5 | | OUT |
| 29 | P1-OUT3 | | OUT | 30 | P1-OUT6 | | OUT |
| Číslo konektoru: | | | K3 | Konektor: | | | TY 513 30 11 |
| Deska/zařízení: | | | JPR-1 | Protikus: | | TX 514 30 13 | |
| Klíčování: | | | C-6 | | | | |

Konečně v článku „Model podmíněného reflexu“ chybí dělicí čára v nákrese plošných spojů, oddělující anodu diody od báze T_3 . V obr. 1 na str. 195 mají být všechny tranzistory T_1 až T_{16} p-n-p, nikoli n-p-n.